

L I S A 08

"An Open Audit of an Open Certification Authority"

De-horning the Dilemmas of Open Trust

Abstract

How does a lightweight community Certification Authority ("CA") engage in the heavyweight world of PKI and secure browsing?

With the introduction of PKI -- Public Key Infrastructure -- as a framework that brought together cryptography, contract law, and institutional views from postal and telecommunications ministries, the Internet security framework rapidly became too complex for individuals and small groups to deal with, and *the Audit* stepped into the gulf to provide a kinder face, in the form of a simple opinion or judgement call. Classically, the audit process oversights a CA for its suitability for reliance in the root lists of popular software distributions.

Yet, a community of Internet enthusiasts does not match the classical target customer of an audit: little money,

loose structures, no deadlines, self-directed tasking, uncertain customer list, all inspired by an original goal of as many free certificates as you can use. Internet communities can make up for an apparent lack of professionalism with enthusiasm, numbers, loyalty and innovative thinking, but does that help or hinder a formal, criteria-directed audit process?

This talk tracks the systems audit of CAcert, an open-membership CA, as a case study in auditing versus the open Internet, community versus professionalism, quality versus enthusiasm. It will walk through the background of "what, why, wherefore an audit," look at how CAcert found itself at this point, and then walk through some big ticket items: risks/liabilities/obligations; assurance and what's in a name; disputes and reliance; privacy and data protection; the mission of a CA; open governance; and systems and security.

Can CAcert deliver on its goal of free certs? The audit is into its 3rd year as of this writing; and remains incomplete. Some parts are going well, and other parts are not; by the end of the year 2008, we should be able to check all of the important areas, or rethink the process completely. Hence, finally, the talk will close with progress and status, and recommendations for the future.

An Open Audit

- Secure browsing and PKI
- Domain of the professional Audit
- CAcert + Mozilla = open audit
- Did it work? Why? Why not?

Presenter

Ian Grigg is Independent Auditor for CAcert. He has spent the last decade designing and building systems of financial cryptography, including payments systems and digital rights and trading systems, with a strong emphasis on secure, open and self-governing systems. Before 1995, he spent a decade as a systems programmer in a wide variety of businesses and roles. He is a frequent commentator on security and financial cryptography issues on the <http://financialcryptography.com/> blog. He has an MBA from London Business School and a BSc(Hons, Computer Science) from UNSW (Australia).

Ian Grigg

- Independent Auditor
- crypto, security, protocols, architect
- Payments, Trading, Governance
- Critic
- Comp Sci + MBA
- *iang at iang dot org*

Admin

- Business briefing paper
http://iang.org/papers/open_audit_lisa.html
- [the slides](#) (in paper in boxes)
- work-in-progress
- [Marcel Simon's talk](#) covered "how"

Introduction

work - in - progress - v1.00

This is a deep and open examination of an audit of an open Certification Authority, CAcert. [1]. It seeks a broad, more open and fuller style as a business briefing, rather than a detailed coverage of technical issues.

All readers should expect to come in for criticism, participants should put their hard hats on. The coverage is open, "warts and all," because the need is to improve, not pat anyone on the back. Many observers of CAcert have come and gone, knowing all is not well, and I think it is futile and insulting to hide the rather spotty past. Better, I propose, to present to the world a message of

"under new management."

Getting better, each day.

1. A Short History of Auditing

In the beginning...

In the beginning, there was nothing: no net, no browser, no root, and no list of roots. Then, one was added. One net, One browser, One root,

And then, another root, and another.

A list of roots! Which created a need for management. Who's roots go on the list? What's the process? What are the

Audits and Openness

- Consider: Why an Audit?
- Assumption: bad news
- Opaque ⇒ No trust ⇒ Establish trust
- Challenge: how to raise the bar?

How Open?

- commitment to openness
- dirty laundry ⇔ open process
- Audit is not PR
- This talk is not "for CAcert"
- hard hats, please!

an Open CA

- CAcert is a Community
 - Assurance: new → Assured → Assurers
 - Business: policy, Board, Arbitration
 - Tech: sysadms, developers, support
- ⇒ Members ⇐

In the beginning...

- Nothing.
- One.
- Another ...
- and another and another

requirements?

- and another and another ...
- Many

For a while, root lists were managed by a more or less informal process that emerged from the market forces. The process involved negotiating with the software vendors (always), paying a fee (sometimes), and being nice (probably).

The CA Systems Audit

"Sometimes when you fill a vacuum, it still sucks."
Rob Pike, referring to something or other

It was into this vacuum that the accounting profession stepped, and created the leading expectation that the one, true, major requirement for being added to the root list was:

the systems audit.

The CA Systems Audit was created by the auditing profession, suggesting that CAs be audited on the same basis as other major IT security systems, a business in which the accounting profession already had a large stake. For example, WebTrust was created by the auditors in the USA and Canada, while ETSI was created by ????

- Need to Manage**
- Vacuum.
 - Systems audit.
 - Pros: fixes known things.
 - Cons: costly, signal.

Although an audit from an accounting firm can add value, it can also cause undue costs, and it can provide mixed or confusing signals. Does an audit signal that a CA is good enough? Good enough for what? And how do we know this?

The audit process seems to have generated more than its fair share of criticism. Salient criticisms include:

- Does not defend the interests of the other parties.
 - Does not make useful statement to relying parties, neither vendors nor end-users.
 - Does not make clear what a certificate claims.
 - Does not make clear what the certificate's signature means.
 - Does not align to vendor practice. E.g., "all CAs are equal" is not reflected.
- The process itself is closed:
 - Is intended for "qualified accountants" only.
 - Parts may be undisclosed.
 - Intellectual property claims suggest that it cannot be used outside the stated commercial context.
- Has not been (properly) updated since its inception.
 - The threat scenario has!

- Cons - Meaning**
- Audit statement? To?
 - Certificate claim? To?
 - Signature meaning? For?
 - Practice.

These criticisms may apply more or less to any of the variants, here they are listed without favour.

- Cons - Closed**
- Nature - secrecy
 - Not updated (?)
 - Threats updated...
 - Bruce S: we are not capable of adjusting our models fast enough.

Mozilla's Dilemma

The informality of the root-list management process was the norm until competition in browsers started up again, around 2003. It worked well enough in the days of no competition, being roughly that period after Internet Explorer destroyed Netscape (the company and the browser) as a viable competitor, up until the arrival, phoenix-like, of Firefox. Where there is no competitive pressure, then the market leader does what it likes, and others follow or not, as they like. No clear guidelines, no strong rationale for one standard over another, and no commonality is required when there is a dominant player.

As Firefox started to show signs of success, questions arose as to how to "get into Firefox." Especially, loud and irritable competitors to

established CAs asked these questions, as the path to their success lay through the root lists. Firefox then was the first port of call in those negotiations, because as the new challenger, it was small, light, nimble and thus the trend setter. Further, as an open product, an open approach to the root list was something that people thought was appropriate.

To Mozilla's eternal credit, these questions were seriously taken. Mozilla appointed Frank Hecker (later CEO of Mozilla Foundation until end 2007) to head a 2 year project to create [a policy for ascension to the](#)

Mozilla's Dilemma

- Mozilla's CA policy
- Existing Audits did what?
- Audits can be open.
- Policy adds: criteria and review (uber-audit)

[root list](#). This process was openly conducted on the public mailing list know as "mozilla-security" and archives will track the full debate. It took 2 years, thousands of emails, and the attention of many people.

The topic was briskly fought. Especially, the question of openness in audits was debated. Many felt that the audit should be conducted to the highest standards, and that was felt to be the WebTrust process established many years ago.

Others challenged this process on many grounds. However, in the event, the debate was resolved by allegations made by the very defenders of the audit process that certain of the WebTrust-audited CAs were conducting activities that no auditor should have approved (they said) and were directly against the interests of Mozilla users, and therefore against the interests of Mozilla.

What these interesting activities were, was never resolved, but rapidly, the group moved to promote a policy of openness, even to the extent of accepting an open and independent audit process. (It should be said, to be fair, that the notion of not relying on an audit at all was not seriously explored.)

CAcert's Dilemma

This extraordinary departure opened the door to allowing open CAs such as CAcert to conduct open audits, but the issue was not solved, only simplified; the audit still needed to be conducted to external and independent criteria.

Enter David Ross, a frequent contributor to the above mentioned debate. As a critic of CAcert, and as a long-term quality engineer, he was encouraged to start an audit of CAcert on the basis of the emerging policy. Ross did not get as far as starting the audit, but he did write out [a new criteria](#) that updated the old, venerable WebTrust version. Ross found other commitments in the way of attendance on the Grand Jury of his domicile.

CAcert's Dilemma

- Audit was still needed
- mid 2005: David Ross: Criteria
- early 2006: Ian Grigg: Auditor

At this point, Ian Grigg then stepped forward and agreed to audit CAcert on the basis of the criteria written by David Ross (henceforth, "DRC"). He, or I, was a long-standing critic of the entire process, so is apparently well versed in at least the weaknesses.

Mission of audit

The initiation of the audit and consequent mission was driven primarily by Mozilla's requirement for an audit to get into the latter's root list for its products (Firefox and Thunderbird).

The process of the audit then was driven by (a) the Mozilla policy, which resulted in (b) the criteria known as DRC (discussed below). The

choices found therein were informed by several circumstances,

- Mozilla's transparency on the root question.
 - Mozilla is an open source organisation, so in principle aligns with other open organisations.
 - Mozilla ran an open security discussion maillist which at least presents a forum of communication on such matters.
 - Mozilla engaged in a strong, open project to create a fair and strong policy for the topic at hand.
- CAcert's founder, the author of the criteria, and the current auditor, were all to be found at one time or another assisting the Mozilla audit project (as archived in the mozilla-security list).
- David Ross's history as a quality engineer, and my history in payment systems and financial cryptography.
- Mozilla's fast growth. At current time musters around 20-30% of the market.

The audit may or may not be applicable or exandable for other purposes. What was still not clear from the origins was what the overall goal of the audit was. After much thought and much experience, the following mission was suggested.

To review and report on the suitability of CAcert's CA to enter the root list of Mozilla Foundation.

However this is somewhat unedifying. In practice the higher goal has been something like:

To review and report on the suitability of CAcert's CA to present certificates to, and provide some utility and protection to, end-users of browsers such as Mozilla's.

That is, unlike other processes, CAcert has specifically moved away from following the flavour of the audit process, and incorporated the end-user as the real customer. This has some ramifications, some trivial and some quite severe, which we shall see soon enough.

Mission in Practice

- *"report on entering Mozilla's root list"*
- not very edifying?
- ***"Do CAcert's certs give utility + protection to end-users?"***

2. Introduction to the Criteria

The criteria, ("**DRC**") are divided into 3 groups or phases, being A (documentation), B (public access), and C (operational review). Within those broad areas there are subsets. Like all tree-structured representations, there are substantial horizontal threads, although these are less well defined.

A full description is best done through reading the source. Here is a summary of the main areas:

David Ross Criteria (DRC)		
DRC reference(s)	Title / Area	Comments
A.1	Configuration-Controlled Specification (CCS)	This is effectively the list of controlled documents that the audit insists is in place. <i>"The configuration-control specification controls controls the revision process for the certificate practice statement (CPS, see A.3)"</i>
A.2-3	Certification Practice Statement and Certificate Policy	The core technical rules of the CA.
A.4	Privacy	
A.5	Security Manual	DRC expects security details to be extracted from CPS/CP.
A.6	Risks, Liabilities	short list of disclosures.
B	Access for Subscribers, and "the General Public"	short list of disclosures.
C.1	Documentation Conformance	<i>"The CA has been repeatedly observed to operate in general conformance with its CPS."</i>

C.2-4	Security, Maintaining Root Certificates	"The root certificate private key is stored secure from electronic and physical compromise."
C.5-8	Generating / Signing / Renewing / Revoking	"Certificates are signed in a timely manner"
C.9	Use of External Registration Authority	RAs are Assurers? "RAs provide the CA with complete documentation on each verified applicant for a certificate (see &A.2,w)"

Comparison with other Criteria

How then does DRC compare to other criteria? In essence, there are these differences to WebTrust:

- DRC identifies a set of controlled documents,
- DRC applies a similar (if not unified) control framework to documents (policies), software, hardware and roots.
- DRC requires the statement by the CA of risks, liabilities and obligations for most of the parties. See below.
- DRC is oriented towards the end-user, and not so much to the CA.

DRC

- imposes control: dox, soft, hard, roots
- risks, liabilities, obligations
- oriented towards end-user, not CA

More to be done?

3. Early Easy Issues

For CAcert, the criteria introduced several big issues, which were in retrospect solved easily.

Configuration-Control Specification

In common with most quality processes the DRC identifies a set of important and therefore controlled documents. DRC-A establishes the Configuration-Control Specification ("CCS") as the set of documents and control policies for those documents that are considered to be so important to the CA's integrity and good functioning, that they form a part of the audit. In other words, the "controlled" set.

From the point of view of the process, this part is simple: identify the document titles, find them, or write them, have them approved (first by the organisation, second by the audit), and make sure they are being used. Simple to say, and it hides a lot of work, but at least this part should have presented no challenge to the average organisation.

History

However, CAcert was no *average* organisation. When CAcert started, it had little of this in place. As of early 2006, the process was that the Board would approve documents, and a policy maillist existed as a place to discuss them. However, few "controlled" sets were identified. no timeframes were established, and indeed no forward plan existed at all.

Things that can go wrong -- The Trap of Being Too Good

An interesting sidebar occurred here. One document that existed was the CPS, or the Certification Practice Statement. This is the core document for a CA, the one that more or less describes everything that should be described. So the presence of the CPS was a good start.

It was however owned copyright by another organisation, and was being used under licence. This sat oddly with the overall sense of "control" as pushed by the CCS. It turned out that the author had (quite valid) concerns about the open source pedigree of CAcert, and had deliberately arranged matters to place the ownership with the Free Software Foundation, and licence the document under their Free Document Licence.

I pondered this and worried about whether the FSF could indeed exert pressure over the CA using this. Although it was unlikely that the FSF

would do this for bad reasons, they might very well try to do it for what they perceived as good reasons. To resolve this, several choices were offered:

- ask for the copyright back,
- audit the Free Document Licence,
- get a legal opinion, or
- rewrite the CPS.

In short, in negotiations, the FSF confirmed that they could and might indeed try to control the document in some sense. They however suggested that we trust them, as they had our best interests at heart. And, no, they were not going to hand the copyright back.

In the end, CAcert elected to rewrite the CPS entirely, from scratch, and this time keep ownership. As the document was well short of audit requirements, anyway, this was the better choice. To be fair, I never looked at the FDL, but I was told it was more complex and less understandable than the famous GPL. Such seems to be a cruel and unusual punishment, and at least beyond what you could ask someone to do in their own free time.

Policy On Policy

It proved fairly easy for the CA to knock out a list to match the requirements imposed by DRC-A. The Privacy Policy was modified with 3 additional clauses, driven by DRC-A, and this was approved by the Board. Slowly, additional documents in the controlled set were created and brought to some state of usability.

However, the process then hit another issue: the policy for approval itself. The Board simply declined to approve anything more complicated than the three clauses added to the Privacy Policy. Documents backed up until crisis struck.

In response to the backlog of policy approvals, it was proposed that this very approval of policy should then migrate to a more consensual style in the spirit of the IETF. In brief, let the entire cycle of policies be done by the community, on the open policy mailing list. From proposal, editing, agreement on draft, and right through to approval of the final document, this could be done on the open subscription mailing list.

To that end, such a policy was drafted as [Policy On Policy](#). This approach reduced the Board problem from Order(n) to Order(1), at least, on paper, assuming it could be approved. The above-mentioned set of backlogged documents could then be handled more quickly, as could the backlog of other policies outside the controlled set.

The process also creates a gap in governance: what do we do with a rogue policy committee. To deal with this, the policy permits two significant escape valves. Firstly, the Board retained a veto over policies that were in the intermediate stage of DRAFT, but the right expired when the policies enter their final concrete status. Secondly, any user can take the policy to the forum of dispute resolution and seek a ruling. The move to *Arbitration* as a means for internal dispute resolution, described later, provides this oversight formerly handled by the Board, albeit in a novel and untested way.

According to its own process, it could be deemed approved. Yet, to deal with the bootstrapping problem or replacing the old approval process, it itself needs to be initially approved by the Board in order for the CA to proceed on its audit path. It is fair to say that the sum of these changes are significant, and in the event, the Board declined to read it, let alone accept it. This became one sign of internal trauma, of which more later.

4. Risks, Liabilities and Obligations

The major area where DRC departs from WebTrust is in the issues of Risks and Liabilities. Consider these Criteria, briefly [here](#), and compare the DRC approach to the WebTrust approach.

Include

**DRC:
R / L
/ O**

- Subscribers -- same
- DRC: *Risks* of the parties.
- DRC: especially, the *end-user*.
- Hard to avoid for ⇔ all parties

../CAcert/Audit/AuditCriteriaOnRisksLiabilitiesObligations.html failed - No such file or directory

DRC Risks and Liabilities are Spread out

Both require the listing of *Obligations* and *Liabilities* of Subscribers (for DRC, A.3.e, A.4.d, A.6.c, A.6.d, B.2.e, B.2.5. For WebTrust, 4, 14).

There are subtle differences: DRC requires the CA to state the *Risks* (being bad things that might happen) of all parties (A.1.h, A.3.j, B.2.c), and especially for the *end-user* (A.6.a). Further, the CA has to state the *Liabilities* that it itself is prepared to take on (A.6.b).

DRC requires that the risks and liabilities of all who come into contact be stated, more or less. The weight of criteria on these points stress the emphasis in a way that it is hard for a CA to avoid; end-users and the public have a right to see it clearly and fairly, and the CA is forced to be open and thoughtful.

WebTrust pushes the Risks and Liabilities away from the CA

WebTrust on the other hand permits the CA to define this quite loosely with its criteria 4,5:

With WebTrust, the Relying Party is expected to carry the load, whereas DRC spreads the load across the three

WebTrust 4,5

1. Any applicable provisions regarding apportionment of liability
2. Financial responsibility, including:
 - Indemnification by relying parties
 - Fiduciary relationships

or four parties, and insists that all the components be documented. In comparison to DRC, WebTrust's rather brutal "indemnification by relying parties" seems to predict it is slanted towards ensuring that the CA is protected from the users. This may be a useful *commercial* objective, but it seems decidedly odd for a systems audit, and its utility to a vendor or end-user has to be suspect.

What is left unstated, and is therefore unclear in WebTrust, is just how these criteria could be interpreted in an audit? Do the said *apportionments*, above, have to be notified? Is it acceptable to bury the *indemnifications* in fine print, or do they have to be advised and accepted by the users? As this is not stated anywhere, the clear incentive is to not do any of it.

WebTrust on RPs

- RPs indemnify the CA ?
- closed provisions?
- What we don't know can't help!
- What we don't know can vary!

Just who or what is a Relying Party?

To underscore this, we can examine the language used, and we find a significant difference. Following PKI convention, WebTrust talks about

the *relying party* whereas DRC talks about the *end-user* and *the general public*. Approximately, these terms refer to people who are not subscribers, rather, those that are given a certificate in order to start an encrypted protocol, check an identity, or check a signature, etc.

This difference in terms is not just a minor issue, it is a major shift in liability, as we will see. *Relying party* is a term that has to be defined, and effectively, we can define a *relying party* to be any person we like. Indeed, this is just to state PKI convention, and the place to define the relying party is the CPS.

stated target?
<ul style="list-style-type: none"> • WebTrust: relying parties • DRC: end-user & general public • definition?

Whereas *end-user* is a term for which we already have a widespread and broad definition in the language. The end-user is the user of general purpose software, such as a browser user, without any special caveats. A limiting definition to *end-user* would stick out baldly and badly. Further, although the term *end-user* might be validly discussed in court, there is no chance of re-defining who *the general public* is, nor of arguing it.

In DRC, the audit will hold the CA to stating the situation for all end-users, a situation that is obviously aligned with the interests of those users, and the browser vendor. In WebTrust, there is all encouragement to ignore the end-user totally, and to limit the liability to those who do chose to become relying parties.

Emphasis
<ul style="list-style-type: none"> • DRC: must state for end-user. • R/L/O + end-user == protect end-user • WebTrust: cover Relying Party? • Indemnification + RP == protect CA

On the Sanity of Dumping of all Liability and Risk

What then goes on in practice? The favourable path is as expected: CAs generally define the relying party to be those who have directly entered into a user agreement with the CA. That agreement then expressly seeks to limit the liability of the CA to zero, and to dump it all on the new relying party [3] [4].

Outrageous, you might say, and this would be the popular criticism! Except, there is one slight issue with this easy cry, and it is a killer issue drawn from simple logic:

Effective Liability
<ul style="list-style-type: none"> • Industry standard: liability to zero • Consider liability of \$1 • across class-action lawsuit (phishing) • Any \$\$\$ is too much!

If the liability of the CA to the end-user, or worse, to the general public, is not contained, then the liability may be incurred by the CA. Multiplied over all end-users, it is simply unbounded. Consider, for example, class-action suits for phishing.

It is then essential to contain this liability in some way. That is, we must have a number, and we must have some risk figure so we can multiply these numbers out and cover the costs.

	How Much Liability?
# of victims:	3.4 million
Average loss:	\$1000
Market share:	0.03% or 100k users
Fudge factor:	10%
Guestimate (1):	$3,400,000 * 1000 * 0.03\% * 10\%$
Guestimate (2):	$100k * 1000 * 3.4\% * 10\%$
Liability:	\$ X00,000

Indeed, it turns out that

Zero liability to the general public is the only sane choice.

By way of counter-example, let's say you want to set the liability to \$1 for each user. According to some measures, this could mean all who saw that certificate, all users of

Zero!
<ul style="list-style-type: none"> • Must contain liability. • \$1 per victim is a lot • ZERO is only sane choice.

Paypal, or, all users of the Internet. In a class action suit, this could mean a buck for *everyone* As there are a billion of them out there, and they keep on growing, this is not what we would call "bounded" in any rational sense.

- (Maybe WebTrust was right?)

You want to offer less than a buck? The only sensible number for liability is zero. The way forward is, no matter how outrageous, to disclaim all liability to the general public and to use whatever legal and technical defences are at hand to set the *expected liability to zero*.

Netscape's Dilemma

Which then leaves us with a dilemma. What good is the CA? Who does it serve? If a certificate gives no protection if it goes wrong, what's the point? If Bob gets phished, how much does the certificate pay out? Zero?

What exactly does a certificate do for the end-user?

And, serious students of business will ask why did Netscape foster the world with this rather odd, asymmetric and

legally-fraught arrangement? Cynics will immediately spot the issue here and say that the CA serves itself, and only itself. (Gee, thanks Netscape!) Well, if CAs manage to disclaim all the liability, to all the people then the cynics might have a good point, Indeed, it may have been this very issue that prompted Mozilla to add in the following clause to the [Mozilla CA policy](#):

1. We will determine which CA certificates are included in software products distributed through mozilla.org, based on the benefits and risks of such inclusion to typical users of those products.

Matched in slightly more ego-centric form by Microsoft [[MRCP](#)]:

Please describe how the services for which your root will be used to provide broad value to Microsoft customers.

...

Roughly speaking, [broad value] means the benefits to including the root certificate outweigh any risks to Microsoft customers. Among potential benefits include alignment of a CA's certificate issuance with Microsoft strategies.

Value to?

- What good is a CA?
- Mozo: "benefits and risks ... to typical users"
- MS: "benefits... outweigh any risks to customers."
- MS (2): "alignment... with Microsoft strategies."

Value #1 - two groups

Group I	Group II
Liability	no-liability
Benefit	<i>no-disadvantage</i>

Value #2 - USE

- use without liability
- like Open Source
- Useful, Free, No Liability.
- defn: "what your software does"

There are ways out of this dilemma for the CA.

- Firstly, it is possible to create two groups of people: those that you are liable to, and those that you are not. As long as the first group gains value, and the second group is not disadvantaged, the benefit to the first group might meet standard of the above stakeholders.
- Secondly, it is possible to deliver useful goods without accepting liability. For example, open source software is commonly delivered for zero price, and zero liability. Open source is still useful.
- Thirdly, it is possible to disclose the real deal, up front, and let users decide. If they still go ahead, they must have found something of use.
- Fourthly, it is reasonable to suggest that the end-user of the target browser is a valid stakeholder in the audit, and consider their needs.

Value #3 - openness

- Disclose the real deal!
- Let users decide.
- If they do: useful :)

Value #4 - mission

- Adjust Mission
- End-user? Member?
- Incorporate their needs.

How CAcert springs from the horns of the Liability Dilemma

Remembering that DRC forces the open disclosure of the full story, CAcert has to take the bull by the horns. CAcert chose these basic tactics:

- Open disclosure of all to all.
- Creation of two clearly separated groups.
- Group 1: **Outsiders**
 - Disclaim all liability to outsiders,
 - but offer USE.
- Group 2: **Insiders**
 - Accept liability to insiders
 - and offer RELIANCE.
- Joining as an insider is free
 - but bound to a user agreement.

CAcert's deal

- Open Disclosure of all to all!
- Two separate groups.
- Joining is free.
- (User Agreement)

Insiders

- Accept liability.
- Permitted to RELY.
- Members

This seems to be a fair offer, although it is somewhat different from what has been imagined. Outsiders can use the certificates, just as is in common with other CAs.

The difference so far is two-fold: Firstly, CAcert will go to the extent possible to tell all users what the story is, and will also attempt to push the appropriate story to others, the non-related persons with whom by definition CAcert has no relationship. Tough challenge!

Outsiders

- Disclaim all liability
- Permitted to USE.
- **Not Permitted to RELY**
- end-users or "Non-Related Persons"

Secondly, instead of attempting to bind the relying parties to a liability of zero, by means of legal devices layered one after the other, CAcert seeks to explicitly set some value in place. As it happens, that value is serious:

€1000

which at the time of writing is around USD \$1446, GBP £790, MXN \$15265, or AUD \$1747.

Liability Deal

- **€ 1000**
- agreement says: from Member
- to whom?

How does CAcert allocate the unbounded liability?

This still leaves a problem. If an insider can claim some value as damages, we still have the same old problem of unbounded liability. Above, we said that any value multiplied by an unbounded user base was unacceptable. As CAcert has unbounded admittability in users, this is potentially the same issue.

In order to avoid the trap of unbounded liability, CAcert simply allocates the liability back to the users. And this "judo trick" is where the user agreement really starts to show its differences.

Liability Allocation

-
- allocated Members ↔ Members
- Judo trick is **Arbitration**
- "Alternate Dispute Resolution"
- own forum, jurisdiction, Arbitrators

Users are not promised coverage of €1000 as implied above. In fact it is entirely the reverse: Users are told that they themselves are liable for that amount. Indeed, by entering into the user agreement, the users are *worsening their apparent liability position from an assumed zero to an amount of €1000.*

In order to accept this, users would presumably have to value the certificates. But what is far more important is how this liability is passed through from one user to another.

Arbitration reverts All to the Users

As with judo, such legal pass-throughs need rules. In order to transfer

the liability in a bounded manner between the Members, CAcert establishes *its own jurisdiction in law*, and binds all users into that jurisdiction.

The technical, legal way of doing this is by means of *Arbitration*, which is sometimes referred to as a form of *Alternate Dispute Resolution* or ADR. CAcert's forum is discussed in a later section.

Summing Up Before the Court of CAcert

In summary, CAcert creates two groups of people, being the Members, and the Non-Related Persons. Where the paths separate for the two classes of people is in the user agreement that either class is faced with. If you as a user have agreed to the [CAcert Community Agreement](#) ("CCA"), you are on the "inside", you are a Member, and you can take your grievances to the forum of Arbitration. If you have not agreed, you are on the outside, and any liability is disclaimed.

Members

- CAcert Community Agreement
- Clause: *You agree to Arbitration*
- Members resolve their disputes themselves
- And can RELY!

Those users who have agreed to the CCA are referred to as Members, and are permitted to RELY on the certificates, which means they can make a financial or otherwise risky decision that requires and is effected by the information in the certificates. If it goes wrong, if for some reason their reliance does not work out, they can sue for damages in Arbitration. In counterpoint, these selfsame insiders are made liable for their own actions, and liability to them is accepted by the CA itself.

End-Users - what about them?

- We don't know them:
"Non-related Persons"
- Take a hint from open source:
 - offer a Licence & Disclaimer
 - Permission to USE
 - No permission to RELY

The second group are offered a licence to USE the certificates. Much like an open source licence, the users of browsers are permitted to use the certificates as they are presented by their software, or by a CAcert Member at the other end of the protocol. What they are not permitted to do is RELY, which means they cannot make a financial or otherwise risky decision that requires the information in the certificate. The disclaimer of liability is simply the flip side of there being no permission to RELY.

How does this work?

- it is the **only** offer
- like "open source" licence
- "Notice" posted in railway station
- CAcert must "post" it prominently

Wheretofore the CA?

Where is CAcert Inc., the legal entity, in all this? The Board? The systems administrators?

Special Parties?

- CA, Board, sysadms
- Auditor: Case a20070921.2
- *"Relief: Mr. Ian Grigg is not allowed to claim to be an assurer until he has 100 points and has passed the appropriate Assurer's test. No penalty is assessed at this time, but the respondent is warned not to repeat the offending behaviour."*
- **All** are equal before Arbitrator

According to Policy on Policy, all policies are binding on all Members and that includes the CA itself. The rules of Arbitration are such a duly approved policy, so CAcert Inc., may appear before the Arbitrator with the same standing as any other Member.

As all volunteers to CAcert are Members, including the Board, the Systems Administrators, and other roles, all parties are before the Arbitrator, equally. Indeed, within a day or so of the policy being

approved, the Auditor as Member was called to Arbitration [5]:

- Case Number: a20070921.2
- Status: Complete
- Claimants: Jens Paul
- Respondents: Ian Grigg
- Complaint:

I, Jens Paul, CACert user, hereby designated Claimant challenge the CACert registered user Ian Grigg for stating that he is an Assurer during the Exec Event. As the CACert system clearly states, Mr. Ian Grigg has less than 100 points and therefore he is not an Assurer and should not be allowed to act as such a person.

- Case Manager: Philipp Gühring
- Arbitrator: Greg Rose
- Date of arbitration: 2007/12/28
- Relief: Mr. Ian Grigg is not allowed to claim to be an assurer until he has 100 points and has passed the appropriate Assurer's test. No penalty is assessed at this time, but the respondent is warned not to repeat the offending behaviour.

There are no exceptions within CACert to these rules, and as the concept of Arbitration is uniformly promulgated throughout the CA, this travels some substantial distance towards meeting the spirit of DRC: *be fair to all*.

The infinite utility of unbounded USE?

What then means *USE*? According to the various policies, it is that which your software does for you, and that which you never do for yourself.

The general public come into contact with certificates in approximately these ways: email, and encrypted SSL web servers. USE works very well for encryption of email, because we simply want to turn on the crypto, and not ask who our counterparty is (us users already know who we are talking to, even if the PKI Industry persists in telling us that we do not). USE also works well for encryption of web server traffic.

USE?

- what your software does
- email, SSL, etc
- Encryption
- Do other checks...
- Not for big ecommerce

However there is one exception: USE falls somewhat short when ecommerce may be involved with non-related persons. This is because ecommerce may involve a RELIANCE by that non-related person, e.g., the act of putting a credit card into an encrypted website. Partly for this reason, CACert does not recommend ecommerce in its CPS, preferring to indicate to Subscribers that, as NRPs are not allowed to rely, exposing credit cards may complicate the position for the Subscriber.

This *offer to USE* is delivered in licence agreement best thought of as the agreement that comes with your open source: you probably do not notice it, but nothing else gives you permission to USE. In that offer, there is also a ban on you RELYING on the certificates, and some other administrative limits.

Members are RELYING parties!

And now we see the final expression of PKI: the Members who have agreed to the CCA are the relying parties. In contract form, CACert is little different from any other CA, however what that contract agreement states is very different.

RELIANCE is then defined to be any decision that you the Member (!) makes on your own. Explicitly, it is not that which

RELY?

- decision you make as Member
- Info in certificate

your software does for you.

- intro in certificate
- disputes => Arbitrator

Actual Value in Liability. Members are explicitly given a liability limit of €1000. They can present their case, and conceivably get an award of damages, although it is entirely up to the Arbitrator to rule on how much to compensate each party, and from where this value is derived. It also means that claimants and respondents themselves are the community, and the balance of fairness for them and everyone else must be sought. What is unusual is that for the first time in CA history, a large group of users have a clear ability to go and get satisfaction.

The Steps in Policy and Documents

This great edifice is constructed of three key policies which are, if laid end-to-end, would probably be shorter than the above entire description.

1. [Non-related persons -- Disclaimer and Licence](#) ("NRP-DAL") gives permission to casual end-users and the general public to USE certificates, but not RELY on them. This is the analogue to an open source licence such as the GPL. Yes, you can use the "software", but only in ways that we say, and don't come crying to us afterwards.
2. The [CACert Community Agreement](#) ("CCA") gives permission to USE and to RELY. But it also binds the Member into Arbitration by means of a special clause.
3. Finally, the [Rules of Dispute Resolution](#) are the procedures that guide the Arbitrator through any given dispute.

Documented Structure

- CCA == Members
- Licence&Disclaimer == end users
- DRP == disputes => Arbitrator
- Principles for soft stuff

x. We do not act to the detriment of NRPs

You may be asked to help NRPs in security. It is your choice to do so, but if you do, you should not act to their detriment. You should encourage NRPs to join the community.

While we work for the benefit of our own users, we must balance our benefit against harm to others. Achieving a benefit to ourselves at the expense of others has no part in our principles.

Other users may join, and they become of us. We exist to help the security of our community, but we also exist to help the security of everyone.

<http://svn.cacert.org/CACert/principles.html>

Finally, a sister document called the [Principles of the CACert Community](#) documents various practices. These are those things CACert Members might desire, but find difficult to turn into a hard rule. For example, the Principles includes a clause that says *"We do not act to the detriment of NRPs."*

Rather than defining such a thing, it is up to an Arbitrator to look at whether this principle is breached.

The Result: A fair deal, and a valuable one!

For perhaps the first time in certificate history, CACert is now offering all its own users a good deal. They can USE and RELY, and if something goes wrong, they have recourse. Further, that has been done without harming the interests of the wider public, unduly. In exchange for the confusion of the past, the general public has clear permission to USE. And, although they cannot RELY, they are permitted to file cases, and can always join as a Member, for free.

Much remains to be done. But what has been done has

Fair

achieved something that is implausible under WebTrust: a good deal for the users. Further, the strength of the basic mechanism means that as a group, security can be the aim, and cash flow can be subordinated to that aim.

- RELIANCE means something
- end-users are not disadvantaged
- doco + resolution
- Free and open.

5. The Management Story

History

In around 2004, the business of CAcert was transferred fully into an Association of Members, registered in NSW, Australia. Included were minuted agreements to transfer the domain names, servers and source code from the founder, Duane Groth, to the Association. A board was duly elected.

These steps created the foundation for CAcert as an independent and professional business. However, a foundation only: the house was yet to be built.

The collapse of the 2004 Board

Above, we touched on the topic of submitting policies to the Board for approval. As well as the lack of progress in policies, there were other issues: lack of feedback on any questions, lack of any minutes or published decisions, apparently years behind on financial reports and AGMs, lack of response to requests for costs.

2004 Board

- Association and Board of 2004
- very quiet in 2006
- Decision: "*disclaim liability to general public!*" "*Or else!*"
- Confusion...

A critical test was reached around mid-2006 when I insisted that the Board approve *the disclaiming of all liability to the general public, "or else."*

Unfortunately, in the panic to avoid unstated sanctions, the Board voted on a garbled proposal forced on them by non-Board members. The result, or whatever could be determined from the email conversations, could be read any way we liked. Literally, it appeared to accept the feared termination of the Audit, rather than their apparent intention to agree with the demands.

In effect, the Board had proven they would simply decline to approve anything, unless nailed over a barrel with roughly-cut splinters, and slowly lowered over shark-infested waters with burning tapers between their toes. Even then, the members would conveniently fail to recall what it was they were asked of, and would yell out agreement in piratical harmony to anything at all: "We agree! Have mercy!"

This left CAcert in the position of grave uncertainty: I had secured some sort of agreement but was unable to work out what the agreement was. I checked with four different Board members and none were able to describe what it was they had agreed to. In detail, I was left with no choice but to proceed with my own directive on the issue at hand. In the wider picture, there was now sufficient evidence that the Board was dysfunctional, and something had to be done.

Increasing the pressure

It was clear by mid-2006 that CAcert has led by a team that had failed to grow with the organisation. The big question was, what to do about it,

The Auditor's Dilemma

- 2006: Board did not approve Policies
- But Audit does not cover Board.

and what part does this play in the Audit?

- *"Management has put in place policies and procedures..."*
- Pressure Point

There is no criteria that says

"management must be competent to consider, agree and approve policies wisely,"

Likewise, the USA auditing profession's *Attest Standard* suggests, not to take on an engagement that cannot be completed, but says little on what happens if one is already engaged. Not so helpful!

However, in the Audit Opinion, being the ultimate work product of the audit, there is phraseology that starts:

Management has put in place policies and procedures...

This gave me the way out of the dilemma. If the management cannot meet this standard, then the Audit cannot conclude. As there was a provable lack of new policies and procedures, it could be asserted that there was no Management. The process of raising the pressure to change the situation then became one of repeating this claim on the mail lists and in other forums of importance. No management, no audit. Logical, reasonable, and while novel, it was sufficient to raise the red flag.

What was less clear to those who cared was what to do about the dramatic claims I made.

And, publically, there were no suggestions offered. An Auditor is not engaged to review the Board, but the CA. Only within the context of the CA and its audit can any strong opinion be made. More particularly, although it might be possible to suggest the Board is not up to the job of managing the CA, it is a very different thing to suggest what it is that the Board should do about the situation.

2006 Failure of Board

- "no management."
- Resignations.
- Loss of quorum ⇒ frozen
- Special General Meeting

Hence, the opinion was couched in terms of "no management," which might have been interpreted as a request to the Board to appoint a manager. Of course, the Board did no such thing as the real and underlying issue was not the absence of managers but the absence of the Board. In time, the absence of all meaningful management activity reached even the slowest and darkest corners of the collective consciousness, and, one by one, the directors resigned.

With the apparent reduction of active directors below 3, around March 2007, the Board entered a state wherein it could no longer form a legal quorum, and therefore could make no decision. The Board was completely frozen, and the onus now shifted to the Association to unfreeze their Board. The members rallied together, called for a Special General Meeting, and voted in a new Board on 25th May 2007.

The Founder Paradox

Success is a lousy teacher. It seduces smart people into thinking they can't lose.
Bill Gates

The root of the issues with CAcert during the years 2004 to 2006 is what is known as the *Founder Paradox*. In brief, one person has a bright idea, makes it happen, and becomes the master of all he surveys. In time, the business grows to the extent that it outgrows the ability of that original founder to see all the aspects. At this point, a team is required; but it does not get put in place because the original founder believes, more or less, that because he got it to where it is now, he knows more than any team.

Students of business know this from their b-school cases on family firms. What is not known is how to deal with it, as there is no good strategy, no 5-points HBR summary to solve the founder problem. It takes years, it's always painful, and many companies fail. Above, in CAcert's case, the answer derived from the earlier 2004 decision to place the intellectual property and the community in the hands of a formal Association, and to elect a proper Board. As this Board failed in due course, and confidence wavered, the Association was still there and was able to rally together to save the day [6].

Founder Paradox

- expert in everything surveyed
- what is not seen?
- Association created in 2004
- formation of new Board

What is also left as the founder's legacy is that once he or she moves aside (as he or she or the business must, in the end), there is a vacuum. This is the founder's legacy. CAcert was no exception to that, and suffered from a predictable and widespread trauma in building up a team and structure to make the organisation work in the aftermath of the loss of its one most productive member.

Luckily, CAcert was an open and popular organisation, and things happened to guide it.

Governance

The Evolving Crisis

As 2006 drew to a close, and as the directors were resigning, matters came to a head in many areas. Action was called for, care was needed. Unfortunately, several issues came up which indicated that the organisation was in more trouble than expected.

Firstly, the servers. As CAcert operated nominally for free, it had no budget to speak of for server costs. Which meant that the servers were colocated at a "friendly business." Now, it needs no great experience to predict what happens in that case, and the business turned unfriendly in due course.

The movement of the servers out of their current location then became critical. Joyfully, a very attractive deal was offered by NLnet, a long-term funder of good works, to finance and manage a rack's worth of equipment in a secure center.

Crisis? What Crisis?

- 2. "friendly hoster" turned unfriendly
- (NLnet suggested Netherlands)
- 3. Audit took time
- (Other CAs suggested subroots ...)

This deal was wonderful, economically, yet complex, legally, and governance-wise. Setting it up took time and attention, and what was perhaps more important, skill. Pressure mounted to secure a deal, and it became fraught. The pressure of time clashed with the needs to secure the operations and maintain a long-term strategy beneficial to the users, and CAcert found itself being pushed into a deal that wasn't good for it.

Secondly, as Mozilla was still "the prize" in many people's minds, and the audit was apparently taking time, alternatives were looked at. The basic concept was to subordinate the CA's root to the roots of other CAs already "in". Via subroots, CAcert certificates would then be represented within the browsers, but for all sorts of business reasons not covered here, the users would also eventually transfer to the new CAs. In short, this is no different to any other insider sellout, except with the possible absence of the customary under-the-table payoff.

The first of these efforts to merge with another CA was a fairly obvious ploy that was spotted by the Founder. The second however was not,

and discussions had advanced to what appeared to be a verbal agreement around a draft Memorandum of Understanding.

Oversight by the Auditor

No experienced manager was in place who could deal with contracts and commitments. When too many of these exceptional cases occurred, I decided in December 2006 to assert direct oversight.

This act was not done lightly. If this were a financial audit, this would be a signal akin to bankruptcy or other failure, and within Australian rules, I would be required to report this event to the financial regulators. The severity of oversight needs to be seen in that context.

Audit takes Control

- no experienced manager
- Need to protect users and directors
- Asserted control December 2006
- Serious signal!

Not only was there an impact on the CA, but also the allocation of liability became difficult to control. Potentially, anyone could then blame the auditor for any misfortune that followed. No matter the rightness or fairness of any such claim, there would be distractions, challenges to independence, and even suits and costs.

On the other hand, there were in the order of 100,000 users to consider, and the wellbeing of the directors and other people on the inside. The actions were creating potential liabilities for the directors who remained at the time, and they were clearly not capable of understanding or handling these responsibilities.

After some thought, I decided to take the following steps:

Suspension of Audit

- Frozen December 2006.
 - Absence of managers
 - moving servers
 - Also, loss of independence.
- No deals! (Except NLnet)
- Not to enter root lists.

1. The audit was henceforth frozen.
2. Auditor takes direct oversight over all "deals" with other parties. In general, all deals were "off", excepting those that I personally supervised. This essentially reduced to the Netherlands Data Center agreement with what was to become Oophaga Foundation.
3. CAcert was to no longer seek to enter any root lists, and was not to engage in discussions over this issue. At a later time, I wrote on Mozilla's bugzilla that express situation and asked them to withdraw the request. Mozilla complied :)
4. This was quoted for two motives:
 - i. Absence of management, as signalled over the preceeding 6 months.
 - ii. The event of moving the servers from Australia to Netherlands via Vienna would probably break audit requirements severely. Pragmatically, however, the need to maintain and move the servers in their unaudited (and possibly unauditable) state overrode the needs for the audit.Motives that were not quoted however included the potential for loss of audit independence.

This was done by signalling in maillists the above points, as and when I felt necessary. This effectively placed the CA *under administration* until the creation and bedding-in of the new Board. This situation was kept up even after the new Board, as there was no guarantee at that time that the new Board would be able to pick up the pieces.

The Advisory

The old "core team had already taken some steps to put in place more structure. This initiative had two components in a matrix form: a horizontal "products" arrangement, and a vertical "department" arrangement. Over the period of 2006, some people had been slotted into this with only limited success, for the usual reasons: Granting a title to someone does not a manager make.

The second attempt was far more successful. As we crossed into

New Management

2007, two other interested people started to put increasing amounts of time into the project (I have chosen not to name names in this document, but for sake of clarification, these are TH and JP). One was able to secure funding for the creation of the new data center, and the other agreed to take on the education responsibilities that had been created due to the audit.

- **Attempt at new "Officers"**
- **Two other managers turned up.**
- **One on the NLnet deal**
- **Other on Education & Orgs.**

What became apparent over the months of the crisis was that these two very experienced people, along with myself, had the experience to make this happen. However, we all had a very big issue: none of us could take on any formal decision-making role or take on any responsibility in the management of CAcert. All for different reasons, it seems, but the result was the same: no responsibility nor any decision could be made.

The Grey Hairs are there for a reason

"Nature abhors a vacuum."
Some physicist.

To solve this dilemma, we took a leaf out of the European book of corporate governance, and created an *Advisory* that had no powers, no say, no responsibilities and no tasks. In essence, we just gave ourselves a title. And started saying things.

Advisory

- How to make structure?
- Creation of "Advisory"
- No power, no say, no responsibilities, no tasks!
- A title, and some words.

In a vacuum, grey hairs generate less friction, less entropy and less lateral temptation. In effect, although we made no decisions, our words, our careful questions, and our complete examples of policies rode above the storm of white-noise normally experienced by such organisations collapsing into chaos.

Along the line of "this is what you could do," CAcert then proceeded to sort itself out:

- Advisory stage-managed the Special General Meeting to elect a new Board. Members of the Association were nagged to be present, plans and timelines were laid out to regimental precision, new Directors was canvassed and chosen before the event, and motions were carefully written to give a new team a fighting chance of taking control. The entire event was a scripted affair, leaving nothing to chance.
- Fortune smiled, and a new Board was elected on 25th May 2007 with the express mandate to "take control." Not however without her little jokes; as the election forgot to state who the new Directors were!
- In parallel, Advisory proceeded to fill out the above-mentioned Officers Structure with keen Assurers, as available, mostly drafted from events. In this way CAcert was joined by Officers in the Events, Press relations, Documentation, and other departments. We disposed of the "products" area, so the structure reverted to a more classical vertical line. Advisory still retained the Human Resources department to itself, for the obvious reasons.
- Advisory carefully and deliberately managed the transition from the old concept of "core team does everything" to "officers do their areas, and the Board does the rest...." This essentially meant the wholesale transfer of power from a small "inside" core team, a legacy of the Founder's time, to a wider group made up of Officers, Board and Advisory.
- In the post-SGM honeymoon period, it became apparent there was simply too much to do. Advisory crafted, drafted, funded and accomodated a plan to fly the three new Directors to Europe and lock them in a room for a week. From concept to meeting was only two months, including the slow European August, no mean feat for an organisation in chaos.

"This is what you could do..."

- SGM ⇒ new Board
- easing the handover
- filling out positions & roles
- core team ⇒ Officers, Board, sys team
- flew Board to Europe for a week

It is fair to say that in the vacuum of the frozen Board, Advisory rode these issues fairly hard. Advisory continued to pick up a lot of areas

while the Board was finding its feet. As the Board ramped up in its capabilities, the Advisory gradually faded into the background. By the time of the full AGM of November 2007, Advisory was no longer active in its previous form, and was replaced by a formal management sub-committee designated by the Board.

Handover

The election of a new Board then presented an opportunity to unwind the drastic step of control by Audit. However it was not without doubt; what would happen if this Board failed? Would they even understand the issues of handover? Should all things be unwound or should only some things be re-started? What of the liability for proper decisions?

Handover

- As board found their feet...
- Advisory faded away.
- Board declared itself "up to speed."
- Audit gave up control!

In the end, I delayed. The Board moved slowly, and did not in the first few months look at anything big or critical outside their limited domain of meetings and procedures.

However, the decision to meet in September, 5 months after the SGM, put the finger on this issue. Hence, this document was first started as a briefing paper for presentation to the Board at the September meeting, and included a delicately placed question as to whether the Board was ready to ask the Auditor to stand down.

The predictable resulted: barely suppressed outrage, much muttering, and "damn your eyes, sir!" Which easily cleared the way for an informal suggestion for a formal decision from the Board to assert that it was "up to speed." And, the Auditor was to revert to more conventional duties. This established a firm date as to when any liability, etc., be capped, and terminated the affair. An unfortunate and difficult period for all involved with CAcert was now over.

And, finally, some Policies!

In a 3 day session, the new Board read through several policies, *line by line*, and approved them all.

- Policy on Policy, which pushed the job away from the Board and over to an "IETF-style" open group.
- Organisation Assurance Policy, which created the groundwork for verifying organisations.
- CAcert Community Agreement, the agreement for *all of us*, with the big scary liability number.
- Non-Related Persons - Disclaimer and Licence, the *fair but free* offer to everyone else.
- Dispute Resolution Policy, to deal with what goes wrong.

'Top'

- Sept 2007 in Pirmasens, Europe
- Policy on Policy ⇒ IETF-style
- Creation of Community: CCA
- Dispute Resolution Policy ⇒ Arbitration

With this massive injection we were back on track, and the documentation requirements for audit were now around 50% complete.

The executive meeting was one of the highlights of the CAcert experience. In a few days, a room full of professional managers cleared through pretty much the entire backlog. I consider this to be to be a testament to these factors:

Missions, Goals, and other B-Speak

Many of the issues that CAcert stumbled over, during the audit and other forces, were traceable back to that old saw: the lack of a good mission. For students of business, this section will be familiar, and might be skipped.

A Highlight!

- experience & professionals.
- Policies ready for last reading.
- Time! Space! No phones!

To be fair, CAcert was not so far off in this department. Two "primary

goals" are boasted on the [website](#).

1. Inclusion into mainstream browsers!
2. To provide a trust mechanism to go with the security aspects of encryption.

Goals here are things that we are doing that are important to us, they are significant, and they have a beginning and an end. Both above goals qualify.

Mission
<ul style="list-style-type: none">• Into the browsers?• Run a CA?• Deliver certs for free?• Secure the World?• Help the Members to Secure themselves?

The problem then is one of completion. What happens when the browsers include CAcert? Do we sit back and congratulate ourselves on a good job? What happens after apocalypse, and the software vendors start distributing opportunistic keys, wherein no CA need apply?

What happens is what the mission says; this is the one thing that defines us, the one thing that we do, and always do. With it, we are who we are, and successful, without it we are failure, we have no other names.

So when the CA job is done, we go back to the mission, and ask "what next?" When a topic comes along that doesn't exactly speak of the CA role, we go to the mission and ask it for its wisdom.

What Missions have been suggested

1. Deliver certs for free.
2. Run a Certification Authority. Get into the browsers.
3. Secure the users. Secure the world.
4. to secure its members access to the net... to provide security services to members... to facilitate the security and privacy of members...
5. to promote awareness and education on computer security through the use of encryption, specifically with the X.509 family of standards.
6. Help the Users to secure themselves.

Of the above, the first two are really goals, and the first drives the second. The third makes CAcert too responsible, and falters on the fairly vague definition of security.

The last is my favourite. Let's take it for a test drive.

Question?	Answer == Mission
Why is CAcert running a CA?	Because ...
Should CAcert hook in as a subroot under another CA?	If ...
Should CAcert advertise other CAs on the website?	When ...
Should the servers move to Europe?	Iff ...

Now answer the question with each of the above missions:

"Because it helps the Users to secure themselves."

If it sounds good for all the difficult questions we can find, then we may have a winner. If not, more thought needed; and try substituting another mission.

Why a Mission?
<ul style="list-style-type: none">• Mission answers questions• Do we do X? Yes if the mission says so.• a <i>work-in-progress</i>• post-Audit

work in progress ...

Where the Mission will help

The mission helps to define things that should be done, and as importantly not done.

As the mission is neutral, it helps to resolve opposing camps, and to keep people honest. If they can show their proposals lead logically to

the mission, without challenge or undue cost, then the proposals are strong, and the people proposing are less important.

It also sets the future. As CAcert moves into a mature CA rollout, it can take pause and think what the mission suggests should be done next.

6. Assurance

Identity and Certs

The (very) basic idea of Certificates

The basic idea of certificates is that it shows your identity to people. The theory is that if your identity is good, you can do some trade or communication with others who also have good identity. The certificate will communicate that good identity, and will also let you use some crypto to further secure your trade or communication. This could either be encryption of your traffic, or making some claim about another document ("I'm selling my house!") or yourself ("Hi Mom, it's me!").

Then, as a basic expectation, all Certification Authorities should:

- Check your identity carefully
- Sign your key with your identity. Carefully,
- Not sign your identity to someone else's key?!?!?

Clearly, this (very) basic idea depends heavily on something called Identity.

Certs: Basic Idea

- Identity is necessary for trade
- Certs show identity
- also: Encryption and Claims

To do Identity

- Check identity
- Sign that identity
- Not sign to someone else.

The PGP idea: Web of Trust

In order to manage your Identity, CAcert adopted the *web-of-trust* model ("WoT"), as pioneered by the PGP community. In a *web-of-trust*, each of us can state what we know of a person, and sign that information so others can reliably get at it. The underlying notion is that we the people have a better grasp of who we are, and a capture of the distributed links of relationship will tell us what is useful to know better than a centralised service can deliver the information.

As an old PGP hacker myself, I have little trouble with the concept, but the details matter [7]. Specifically, the PGP *web-of-trust* was fine as long as nobody relied upon it. And, even then, as long as each individual understood that the information was raw opinion from others like themselves, and the reliance was on an *as is* basis, it worked. Of course, these claims apply to most every design, including the competing notions of PKI.

Web-of-trust

- Everyone can make a statement
- assumption: I know more than you
- assumption: we know more than one person
- limited by assumptions!

The CAcert Network of Assurers

CAcert adopted its WoT further by limiting the statements of people's Identity to people they call *Assurers*. This was possibly adopted from Thawte's "notary" model. Each Assurer can give up to 35 points, and as you need 50 points to get a Named certificate, this implies checks by two independent people. The Assurers were defined as people with 100 points, thus implying that they were vetted by at least 3 other Assurers.

CAcert's WoT

- Only Assurers make statements
- Name statement: 2 checks, 50 points
- Assurer: 3 checks, 100 points

CAcert's Assurance network consisted of around 10,000 people, checking over a base of some 100,000 people.

Assurers

- 10,000 Assurers, 100,000 Members

At its simplest level, the Assurance process is a careful human face-to-face verification of government-issued photo-Identity documents.

- face-to-face
- Govt. issue Photo ID
- Exceptions: TTP

raise your hand if you are an Assurer

The PKI Idea: the Registration Authority

In contrast to the above notions, the PKI idea was grounded firmly in an authority for every important act. This is mostly traceable back to the golden age of telecommunications whereby national champions had a licence from a government, generally one only per country, to control the communications of a country. In that world, it was an article of faith that the answer to any question could be found in the designation, or creation, of an authority. Preferably a single, national one, but always one with gravity.

PKI's Registration Authority

- assumption: there must be an authority
- assumption: the authority is singular
- limited by assumptions!

In the PKI literature, a single *Registration Authority* is generally required to verify the user's Identity and vouch that to the CA. From a helicopter view, we can see that the differences are much smaller than talked about. From the anarchy of *web-of-trust*, CAcert reduced the statements to a few thousand Assurers, requiring two of these to agree. From the superficial authoritarianism of the *Registration Authority* model, recent commercial models pushed the function out to thousands of external companies, and recent audit designs now require the process to be conducted by two or more individuals within an organisation. In this way, the differences between CAcert's system and the PKI model are really at the detailed levels of quality, and not at any fundamental level of design and structure.

Difference?

- PGP Anarchy ⇒ Assurers
- PKI Autocracy ⇒ RAs
- Diff #1: Numbers?
- Diff #2: Standards?

Flaws with the Assurance Process

In the event, DG put in place a relatively good system, so only minor tweaking was needed. We can attack this concept at three levels, being *what*, *how* and *why*.

Auditing thoughts...

1. what is the *standard*?
2. how is it *Checked*?
3. why is it *useful*?

First the standard. It seems to be clear that the standard of your Identity is critically dependent on the documents that attest to your Identity. In this context, CAcert had more or less decided to go for government-issued, photo identity documents by the time this audit had started. While this standard has many, deep, political, philosophical and geographical issues, *it doesn't really get any better*. To cut short a long story, this standard was accepted without change.

Second, how is it checked? CAcert had already decided to check the identity of the users with the users themselves, as decided above. As described, at least two Assurers would check the identity before it could be put in your certificate, and three at least were required before you could also check others' identity.

Who Assures the Assurers?

Quis custodiet ipsos custodes?
Decimus Iunius Iuvenalis, a.k.a. Juvenal

Which led to a several problems.

- At a simplistic level, how do we know who the Assurer was?
- Or, in the extreme, who was *the very first Assurer*? Who was the second?

Checking

- Are the Assurers doing it?
- Is the check consistent?
- Who Assures the Assurers?
- Can I attack it?

- At a fairly simplistic level, how do we know that anyone is checking?
- At a more sophisticated level of quality, how do we know that the check that is done is good?
- At the highly sophisticated level of Internet security in the world we now live in, what is to stop me entering the network as a nice guy and selling my checks to the highest bidder?

CAcert had few answers to these questions, but with some prodding, several responses evolved.

- The existing network was accepted, as:
 - it had been in operation for many years,
 - few problems had surfaced during that time, and
 - the incumbents had little incentive to pervert the network.
- Checking, or *verification*, needs to be the subject of objective and testable standards, so we need:
 - A standard.
 - A test.
 - A recovery mechanism.

Such things did not exist, although some things did fill their places. A wiki with a lot of ideas and procedures served as documentation, and testing was more done by the *practical but informal* check of experienced Assurers working with newer Assurers. However, these were unreliable. Firstly, there had not been a solid grounding on what Assurance should be, and the wiki varied in its treatment. Secondly, not only was any checking subject to obvious variation, it also tended to encourage the variations to bed in and become divergent standards, often on regional bases. Thirdly, there was a belief that the Assurers were better able to judge everything themselves, following on from the *web-of-trust* school of thought.

Challenge for Assurance
add to current system:
<ul style="list-style-type: none"> • standard: the Assurance Policy • test: the <i>Challenge</i> • recovery: Arbitration

Education

In order to escape from the maelstrom of variable quality, discussion inevitably spiralled into a need for a test. This would be a basic benchmark which could later be changed and tuned as experience develops. Once the concept of testing the Assurers was accepted, the thoughts of the techies immediately turned to creating an online test suite.

Assurer Challenge
<ul style="list-style-type: none"> • Built by Jens Paul and volunteer programmers • CATS went live 2008 • 40 questions, 80% passmark • 1,197 Assurers (06.nov.08)

Luckily, an experienced leader of real business projects had turned up at the time, and agreed to take on the challenge. CATS or *CAcert Automated Testing System* was created over the year 2007 with volunteer programmers, and opened up for general testing of Assurers, beginning 2008. By November 2008, 1200 Assurers had attempted and passed what had now been christened as:

[The Assurer Challenge](#)

Have you passed the Assurer Challenge yet?

The test is a series of around 40 questions. The pass mark is 80%, and a candidate can take it as many times as liked. In essence, as well as a benchmark, the test has a learning effect.

Unusually, CATS was built and run outside the main CAcert environment, so there is a communication required between CATS and CAcert.

Details on CATS
<ul style="list-style-type: none"> • outside the critical systems • Relies on certificates • Valuable learning experience!

After much discussion around privacy and security, client certificates are required to run the test, and the result is communicated by client certificate number; CATS itself does not need to know who you are, simply that you are a Member, and your membership can be uniquely identified to CAcert with the positive result. This was a unique and valuable chance for the whole organisation -- Assurers and managers - - to learn about client certificates.

Have you passed the Assurer Challenge yet?

- "old" "Candidate" Assurers will be turned off
- cats.cacert.org

Unchallenged Assurers will lose their status, and while the code is written to enforce that, it remains for the switch to be thrown. This means in effect that the body of Assurers will shrink from some 10,000 to something under 2000. As CAcert has been collecting these names for some 6 years now, it is reasonable to expect a dramatic shrinkage, as well as professionally responsible to keep the status of the data up to date.

The Assurance Policy

As mentioned above, there was no solid grounding in what an Assurance was. Indeed, there had even been a rule that "*Assurers can do anything*", and occasionally this rule was taken to heart by overzealous Assurers who were sure of their own realities.

Unwinding that rule took a long time. The first nail in the coffin was liability, as most of the Assurers equated *freedom of action* with *zero liability*. However, courts would be unlikely to agree to that. Luckily, the crafting of the risks, liabilities and obligations project resulted in the creation of the CAcert Community Agreement which both formalised the liability of each Member, collected it before CAcert's own forum of Arbitration, and put a limit on it.

Standard of Assurance

- "*Assurers can do anything*"
- Assurance Policy
- splits out detail to Handbook
- now in DRAFT: binding
- (slow progress on Policy Group)

Then, the project to build and impose the Assurer Challenge woke Assurers up, to the extent that they knew it.

The final requirement then was an objective policy against which the test could be written. This was the first major work of the newly empowered *policy group*. As its major act of 2007, the Board had passed the policy responsibility almost entirely to the open mail list, and the Assurance Policy was their first challenge.

Progress was slow. In practice, each section had to be argued over, and fights erupted over seemingly trivial details. As one detail would appear important to one person, and trivial to another, there must have been some merit in it. Finally, after some 6 months of discussion, the Assurance Policy entered into DRAFT.

The Assurance Policy has a number of notable features. Firstly, it sets a general standard for the Assurance of individuals. Specifically, the Name is defined, the documents identified, the basic check is laid out, and the points system is established. This finally establishes a benchline for what had been argued in the community for a long time.

Secondly, although it is a firm policy and hard to change, it explicitly splits out a lot of detail into something called the Assurer's Handbook. This is a dynamic wiki document, nominally under control of an Officer. It is argued that because the threats move too quickly for policy deliberations, then much of the day-to-day practice will be better dealt with in a document that is not so heavily controlled by audit provisions.

Thirdly, it establishes the definitions of Assurers and gives authority for the Challenge.

What's in a Name?

*What's in a name? That which we call a rose
By any other name would smell as sweet*
William Shakespeare

Above, we stated fairly baldly that the point of a certificate was to provide your Identity, which might be simply seen as recording, verifying and presenting your name in a certificate. E.g., a recent document said [8]:

Classical PKI model

- certs provide names
- Before: trading with good people
- After: chase the bad person

The primary purposes [are]: (1) Identify the legal entity that controls a website. Provide a reasonable assurance to the user of an Internet browser that the website the user is accessing is controlled by a specific legal entity identified

And further:

The secondary purposes [are] to help establish the legitimacy of a business claiming to operate a website, and to provide a vehicle that can be used to assist in addressing problems related to phishing and other forms of online identity fraud. By providing more reliable third-party verified identity and address information regarding the owner of a website, [they] may help to:

- (1) Make it more difficult to mount phishing and other online identity fraud attacks using SSL certificates;
- (2) Assist companies that may be the target of phishing attacks or online identity fraud by providing them with a tool to better identify themselves and their legitimate websites to users; and
- (3) Assist law enforcement in investigations of phishing and other online identity fraud, including where appropriate, contacting, investigating, or taking legal action against the Subject.

That is fairly typical of the thinking in the CA industry. Traditionally, the claim made to sell certificates was made in two parts:

- Before: You need the name to trade with the person [9].
- After: You need the name to chase the person when it goes wrong.

Reality of the Internet

- Seller needs money
- buyer needs goods
- Before: eBay + Reputation + Credit?
- After: Reputation + Resolution

But this is just plain wrong. We do not need the name to trade, the seller just needs the money (therefore any credit card is fine) and the buyer needs the goods (therefore any address is fine). With or without the name, these things work fine, and eBay's reputation system seems good evidence of this. The same goes for most other aspects on the net, a good name for a person is simply not needed, although it is more polite than a good number.

Afterwards, we might need to chase the person, but the reason is that we want to see them in court. However, getting someone to court is only helped marginally by having the name. What helps more than anything is proximity and reputation. Sadly, the utility of all these things go down, as we get out and on to the net, and names in certificates suffer exactly the same fate as everything else on the net: it is harder to dispute as the distance increases, and transaction costs will force most low value, long distance disputes to be un-heard.

Resolution

- Depends on cost of lawsuits
- (Cost goes up with distance)
- Depends on value of trade
- (transaction costs hole)

For example, in the case of "Eros LLC vs John Doe (Tampa)," an Avatar, Volkov Catteneo was pursued and found in real-life [10]. The overriding factors were the location of the person in Texas, the ability for the system of dispute resolution to reach

What's in a Name?

*That which we call a rose
By any other name would smell as sweet*

- Name does not decrease distance
- Any name will do
- fashion accessories: Volkov Catteneo

that person, and the desire of the pursuer to chase the person that far. The name was not that important.

The migration from Identity to Community

However, perhaps we can skip the the utility of the name in helping us to get to court, and just look at the forum itself. Classical courts are too expensive over distance, so what alternatives exist? It turns out, quite a lot, and further, CAcert has already put in place a mechanism, its own forum for Arbitration. For that, we turn to the next section.

Can we Resolve?

- CAcert's Arbitration!
- Reaches all Members
- reduces distance
- Costs very low

7. Arbitration

You can't punish a key. What would you propose doing? Lop a bit off?

Steve Kent

As seen [above](#), Audit forced on CAcert a clear exposition of *Risks, Liabilities and Obligations*, which in turn faced it up to a need to control liabilities, both

internally and externally. Externally, liabilities were controlled by disclaiming them, but internally it was felt that in true PKI style, Members should be able to RELY on certificates.

Arbitration

- Derived from liabilities
- Dispute Resolution Policy & rules
- Board, AGM, and policy group
- Nov 2008: 7 complete, 1 pending

How then was this reliance to be controlled? How were the consequent liabilities to be disputed, allocated, and recovered? Any court case would be too expensive for any Member to think useful, which led CAcert to look at *Alternate Dispute Resolution*. In the event, it was decided to put in place binding Arbitration amongst Members. A set of rules was written, and at the September 2007 meeting of the Board, the policy was approved [11]. "Out of an abundance of caution," this and other policies were also presented before the Annual General Meeting of the Association, and to the newly-empowered policy group. As of November 2008, seven cases have been dealt with and one is pending [12].

Some Classical Advantages

Arbitration is a method of dispute resolution that is conducted outside of the normal courts, but is backed up by the normal law which those same courts work to. This law is commonly called *The Arbitration Act*, an Act passed in most countries [13]. The import of this is that courts are generally at peace with the practice of Arbitration. Indeed, courts will refer cases to Arbitration, and do so happily.

The Arbitration Act

- Most countries
- USA: The Federal Arbitration Act
- Courts refer cases where appropriate

The strength of the Arbitration system in CAcert derives from the Arbitration Act found in most countries. Here is a brief description of the advantages for CAcert:

- Arbitration can be tuned to the needs of the community, the costs can be aligned to the nature of the disputes, and the experience needed can be relied upon to be available. CAcert

Strengths

- Tuned. Expertise. Cheap.
- other roles

appoints senior and experienced people in as Arbitrators, which helps to maintain high expertise in technology, and reduce costs.

- Arbitration can be turned to fill other roles, outside strict "disputes." For example, CAcert can consider a lost password, a phishing attack, or the data protection issues behind an account closing in the same forum.
- The CA can avoid costly and fraught court cases, because (within some limits) the courts will refer a dispute back to our Arbitration, citing that country's Arbitration Act.
- The forum can choose for itself the law, and the rules. These can be applied *universally* and *equivalently* to all of the community, which to a large extent will protect individuals from strange and sometimes dramatically harsh treatment in a strange and faraway jurisdiction.
- It is possible to invite non-related persons in, and give them some satisfaction, again, within manageable bounds.
- It avoids the "rules-based" approach to security so prevalent in documentation and quality approaches today. Instead, rare conditions are referred to an independent but experienced party to make a determination. This reduces the documentation and policy substantially by replacing large and complex rules with the three most valuable words in the CAcert book: *file a dispute*.

- Avoid cost of court
- one law, forum *for everyone*
- Non-related Persons can enter
- fewer rules: *file a dispute!*

What holds up the Rose?

Above, we saw a claim that perhaps the real benefit that users want behind certificates is to get their counterparty into court. Arbitration can be seen as a mighty fine substitute to the more conventional courts, and in this it answers to that real need.

The original Assurance Programme has always provided a pretty good basis for verifying the Member's Name. And in its improved version, with a policy, formal Handbook and tested Assurers, it provides a good *and solid* basis.

Grasp the Stem

- Arbitration gets Member to forum
- Assurance now looks to wider issues:
 - is a Member, can Arbitrate
- Result *of all* is useful
- Name is detuned
- Still useful, no longer critical

But the true strength of Assurance is now found elsewhere than the conventional check on Identity documents. A CAcert certificate now makes five important claims about a holder, taken from the Assurance Policy [14]. They are, briefly:

1. That the holder is a Member,
2. That the Member has an online account,
3. The certificate identifies the account and Member,
4. The Member can be brought to Arbitration,
5. Some details such as the Name are known.

It is the 4th one that is key here: With the Binding Arbitration clause of the CAcert Community Agreement, the Member has agreed to stand before a peer as *respondent*, as will the initiator of the dispute, *claimant*. Typically, many of the systems and policies of the CA are now oriented towards this goal.

How this happens

In a hopefully fair and open manner, the CAcert Arbitrator will hear the dispute, and deliver a binding ruling. At the Arbitrator's disposal is a list of remedies, from mild to severe: from community work days and loss of status (points) up to €1000 of fines and ultimately ejection.

Accepting all the above, we can finally get to the end of this long path to understand the big shift in Assurance: the Name within the certificate is no longer (as) important for reliance, as CAcert has other methods to encourage Members to behave. To some extent, the Name is now reduced to a cosmetic issue, and this matches the reality of the net far better. In the general Internet use-cases we know of, it is probably not necessary for you or your counterparty to have a detailed, verified Name, but an Assured Member can certainly put it in there if you want.

(As an exercise, the reader may like to work through what happens

when disputing a certificate holder without any strong name in it, whether absent or a nickname.)

For Everyone Else Out There

There are limits. An important one is that the possibility of remedies by Arbitration is not necessarily available to all. Let's examine that. The offer made to Internet users states that the person is **not permitted to rely on the certificate** [NRP-DaL]. Yet, the dispute resolution rules clearly permit that anyone may file a dispute [DRP].

Everyone Else?

- NRPs not permitted to RELY!
- NRPs can file a dispute?
- no monetary remedies
- punishment still possible

Any parties that are not Users and are not bound by the CPS are given the opportunity to enter into CAcert and be bound by the CPS and these rules of arbitration. If these Non-Related Persons (NRPs) remain outside, their rights and remedies under CAcert's policies and forum are strictly limited to that specified in the Non-Related Persons -- Disclaimer and Licence. NRPs may proceed with Arbitration subject to preliminary orders of the Arbitrator.

Superficially, these two provisions seem to be a contradiction, but actually they work together. Although any non-related person can file a dispute into the CAcert forum against a certificate holder, the Arbitrator is not likely to give that person any monetary compensation. If they are not permitted to rely, then, and have still got themselves into harm's way, then they have taken matters out of CAcert's hands and into their own.

Yet, the absence of compensation to the person does not mean the absence of punishment to the Member. Indeed quite the reverse is possible, and the Member could find himself punished according to a list of remedies listed in the rules. Some are benign or soft, such as a day's service to the community or the loss of status (points), but the Arbitrator has the power to fine up to € 1000 or to eject the person from the Community.

Other Incentives of Great Benefit to Audit

There are other incentives which assist Audit immeasurably, and explaining them may help to understand our enthusiasm for the project. As well as the above roles, it can also, as a process and procedure reach into much broader areas: governance, support and diplomacy.

Arbitration as a method of Dual Control. Consider support actions to fix a "lost password" as a trivial case of dual control. It would work this way:

1. Bob, a user, mails support to get a new password.
2. Alice, the support operator on the day, collects all the information from the user, in the normal way.
3. Once she is convinced that she has correctly identified the user, as Bob, she can "file a dispute" on his behalf, against herself.
4. Trent, the duty Arbitrator, is then allocated directly to the case. He calls for evidence.
5. Trent reads through that which Alice presents, as evidence, and writes and publishes a ruling which instructs the operator to change the password.
6. Alice follows the ruling and changes the password. In the appropriate support form, she enters in the ruling date and index number into the field that asks for her authorisation.
7. Later on, an audit process scans through all of the password changes, and matches them up to the rulings.

Obviously this process is more complex than the routine act of a support person changing passwords common with user-facing systems. But CAcert online accounts can issue certificates, so changing passwords needs a bit more care. The complexities more or less derive from the requirement for dual control, not from the usage of dispute resolution as a support mechanism.

(Note that this is a hypothetical, CAcert does not currently

do this.)

Hence, where there is a requirement for dual control, Arbitration can slot in to fill that need.

All Arbitration rulings are by default public. CAcert Members have an incentive for CAcert to be seen to be good, not bad. Further, the principles of the community specifically state, as mentioned above, that *We do not act to the detriment of NRPs*. These forces encourage a fair amount of interest in the Arbitration project, and seek to keep it working to the benefit of all.

Audit Benefits

- Dual Control
- Rulings are published
- Criminal: Arbitrator
 - authorises
 - documents
- post-Emergency authorisation

One exception has to be noted: Arbitration is about *civil disputes* and not *criminal cases*. In the event of prosecution, a judge will not refer such a case over to Arbitration; but it should be noted that pretty much all of the analysis holds, and CAcert remains in a better position in criminal cases with Arbitration than without.

What then happens if a civil case in Arbitration drifts into criminal matters? Likely, a case will move along and document its findings. A civil case is not a trial, and provides no real protection against a prosecution of the Member. On the other hand, any ruling will be of interest to a court. The ruling should provide greater certainty, and a lesser cost, even to the extent that it may end up contributing much to the expert needs of the later forum. This is a positive for the Community.

What about the other way around: a prosecutor launches the proverbial legal strike against CAcert or the Members? Again, the Arbitrator steps in to help. If the court's order is for information, this cannot be provided in the general case because nobody has authority to provide anything but public information. Hence, anyone subject to a court order is required to file a dispute to get the authority. This becomes an essential dual control on all secrets, helping to keep them safe, and only revealed under proper circumstances.

Likewise, the process for emergency actions and breaches (where the latter might be by any means) benefits. According to the principles of governance, Administrators are not permitted to make changes without controls in place, and that includes installing new patches that might stop attacks. Bad things sometimes happen, and when they do, the rule is that the administrator must act according to best judgement, but must then *immediately file dispute against self*. The dispute before the Arbitrator then leads to the Arbitrator reviewing the actions, likely delivering a post-event authorisation, and possibly delivering further guidance or even sanctions where events were not well handled.

In practice, then, although the person out on the Internet is *not permitted to rely on the certificate*, there are substantial and solid reasons why a CAcert certificate can be considered to be a reliable thing.

Roles and Uses

Over time, CAcert's dispute resolution has evolved to fill these roles, some of which are written in as policy or practice, and some of which have evolved into actual cases:

- *The original motive*. Resolution

Benefits for All

- allocation of Liabilities
- Hearing for browser users
- Dual control
- Training and Teaching
- Knowledgebase of Rulings
- A bridge: Tech ⇔ Law
- Unified across distances, reduction in proximity
- Protection for Members, for CA

of disputes surrounding relying party actions related to certificates. E.g., Member-to-Member certificate issues.

- Sharing of power
- Feedback, reflection on policies
- Representative of all

- An offered response to disputes from external parties, within the softer scope of the wider browser user. That is, we may disclaim liability to the browser user, but we do not discourage a fair hearing, and this may still lead to a sanction against a Member.
- A Dual Control method over critical support actions such as the revealing of privacy information.
- A way to ignore defer tricky cases from slavish documentation. Where a policy becomes fraught, it can cut away the complications and terminate simply with "file a dispute." A live, experienced human reviews and rules on the case.
- A training exercise, a teacher of the rules and ways of CAcert.
- A way to build up knowledge, by means of the records of the cases (primarily, Rulings).
- A bridge between technology and the law, a way to bring them together comfortably.
- A unifying layer for all Members, especially effective in smoothing out the diverse and often arbitrary laws found in some jurisdictions over matters of technology and security.
- A way to protect Members against arbitrary use of civil procedure.
- A bulwark against liability for civil actions, achieving something akin to an insurance policy.
- A powerful tool in power-sharing. As Arbitration has the power to strike down or rewrite Management actions and Policy approvals, this protects against either of those two bodies from getting too powerful or out of control.
- A reflective body that allows the principles and missions of the Members to surface and impact the operations.
- A CAcert Arbitrator is a representative and diplomat for the entire body community; and has to stretch to draw all interests in.

With so many advantages, it is no stretch to see that the dispute resolution system pays for itself in its flexibility. At the time of writing, there have been a handful of disputes on the issue of *What's in a Name?*, and now one serious dispute on the issue of the closure of an account. The issues were complex, but the Arbitrator surmounted them, and ruled. Others can now follow that ruling and build upon it.

Criticisms of Arbitration

There are many **disadvantages** and these are widely discussed in legal circles. Here are some of the controversies experienced to date:

- As paraphrased by a learned friend of mine, "*Conducting Law without a licence!*" [Sir Edward Coke] Although somewhat surprising, it is intentional. The experts in the subject matter are the senior Assurers in the CA, and the rules are not so much the law, but the policies. Indeed, the Arbitration Act generally encourages the deferral of specialist cases to bodies better informed of their peculiarities, albeit still under the general rubik of the law.
- As Arbitrators are chosen from within the community, they are not necessarily free of bias. This is probably reasonable for internal disputes, but could be reasonably questioned for disputes with those who are not Members. A future dispute with a third party vendor, for example, might seek a more independent panel.
- *No lawyers!* Again, this is intentional. The disputes within CAcert are low-value, procedural and generally non-antagonistic. Reasonable people can disagree on simple questions such as
 - does the Member have the right to see an Assurer's status,
 - whether a middle name is J or John, or
 - can an ex-Member demand the removal of all details?

Lawyers are not going to help with such community issues, or at least, they can only do so at high cost. It should be possible for the subject expertise and the professionalism of the insiders to offset the lack of legal training, and to date there has been no evidence to the contrary.

- No training, no professional guild. This is more a reflection of youth. The only way forward is forward, CAcert needs to get a few cases done, and follow a natural progression.

Criticisms

- Avoiding the Law? Rules & Law
- Bias? must be open and careful
- Lawyers? expertise & value
- Training? time, future issue
- Criminal? still via Arbitrator

The Role of Audit in the Question of Arbitration

As we saw above, Arbitration changes the character of Assurance quite a bit. More importantly for this author, it makes questions of auditing the quality of Assurance, and most other areas, much more tractable. Before, auditors had to ponder the imponderable: what identity

document is better, are 3 middle names too many, and how do we deal with those foreign characters?

Now, Audit can look at whether the system leads to a process of resolution of disputes, and that, once filed, the process leads to a just, fair, efficient, and open result in any hypothetical problem at hand.

Obviously, there are details of how we travel from Membership and Assurance to reliance and usage, and thence to Arbitration and a ruling; these details are not trivial. They might be expensive, in time and resource. There can still be problems in any given case.

Yet, for all these costs, CAcert has surmounted the barriers, has conducted a handful of Arbitrations, and has delivered results, all of which has served the Community well. Arbitration is workable, it is efficient, and complete. The system applies across broad swathes of policy and practice, making the audit process much simpler. Even better, it goes further in reducing the role of the Auditor, as *any Community Member can review the rulings*. That which is open and is verifiable by the public does not need to be audited.

Perhaps best of all, it has finally given the certificate some sense of the value and pride, promised to it.

Grasp the Stem

In summary, CAcert has therefore established:

- The certificate makes a claim: that the holder is *part of the Community*.
- The claim made by the certificate is backed: *anyone can file a dispute against that person*.
- The claim itself is accurate and not deceptive -- if disputes can be filed.
- The claim is useful -- if filed disputes result in a fair hearing and a fair ruling.
- At least, the claim is as useful as any other is made.
- The process is cheap. It might or might not be free (fees are permitted under the policy but not currently charged)
- The process (or its costs) is aligned with potential areas of disputes rather than against (as happens with the courts). Any disputes over certs are probably low, and any large disputes should involve the time of volunteers who follow the incentive to maintain good reputation of the Community.
- It creates its own jurisdiction, and therefore works without additional boundaries (and more costs) for all Members wherever they are. Proximity is created for all Members, the tyranny of distance is gone.

Is this the first time ever that a CA has done something that is both robust and potentially useful with certificates? Perhaps this might be an exaggeration, but it does seem that the CA industry's 14 year obsession with the One True Name has been taking us down the garden path. Instead of sniffing roses, we should be grasping stems: methods to hold subscribers to account are neither difficult nor expensive, and giving something of value and substance to relying parties might make certs more popular.

This might or might not be the first time that a certificate's promise is backed up with something of substance, and is not deceptive, however, barriers still remain: we still have to show that certificates can do more of service than act as gatekeeper to crypto.

8. Organisations

Everything in the foregoing concerns *Individuals*, and it is fair to say that the system of individual Assurance was reasonable to

Results for Audit

- Dual Control
 - post-Emergency authorisation
 - demands
 - breaches in policy
- read the *published* ruling
- complexity can be deferred: less doco
- Community can govern itself

Grasping the Stem

- Claim: of Community
- Backed: file a dispute
- Accurate, not deceptive, useful, cheap
- Aligned: disputes, costs, net, proximity

begin with, and is now stronger.

Not so with Organisations. For reasons that we can only speculate on, Organisations that are assured in CAcert have never looked as sweet.

<p>Organisation Assurance</p> <ul style="list-style-type: none"> • poor doco • unresolved conflicts in process • will likely miss the boat

Situation in 2006

When the audit first started up, these were the issues:

- Organisation Assurance was not documented.
- The standard at the time was by reference to local law. This might sound promising, but problems occurred:
 - It dramatically reduced the applicable number of organisations.
 - Especially, it discriminated against many of CAcert's supporters who operated small self-owned businesses.
 - It created a standard that was at odds with the philosophy and style of CAcert.
 - The standard of one country was thought to be good for all countries.
 - Which created a trust war as different cultures fought to export their view of trust to other areas.

These things derived from a false assumption that *only* the standard in legal registrations was good and applicable, with no thought as to the many and varying circumstances in the wider world. The CA's own requirements were thought irrelevant, and the people who were disadvantaged could not "compete with the law." This quickly become ludicrous, because the one country that championed this approach was considered over-strict by its neighbours, thus leading to a monopoly on issuances.

<p>Dis-Organisations</p> <ul style="list-style-type: none"> ◦ no doco ◦ reference to local law <ul style="list-style-type: none"> ▪ discrimination ▪ principles ▪ export ◦ reserved to specialists
--

- The method of verification of Organisations was reserved to those with juridical training. Again, this might sound good, but in practice it reduced the number of people available, and it allowed an *appeal to authority*. In the end, it became a way to assert the above political view that one country's regime of registrations was good enough for all, and an easy excuse for no standards nor training.

Even as audit was looking into the basic form of Individual Assurance, the war was brewing. Within the one leading country, trouble erupted when some people discovered that certain forms of organisations were to be completely rejected because the form of proof was not up to the standards proposed by the people running Organisation Assurance, and there was no intention to change that.

Advisory's Policy

At CeBIT of March 2007, enough evidence was presented to conclude that, at the least, the Board had no control over the process. To address this, I stepped in and put a stop to all Organisation Assurance until a policy was developed and placed into at least DRAFT. Advisory got together one weekend and knocked up a working draft of an Organisation Assurance Policy [15]. This was approved by the new Board as its first major policy at the September 2007 meeting.

The new Organisation Assurance Policy (OAP) was a reasonably good effort. It had one endearing feature: it split off all the different forms of organisation into what are called Subsidiary Policies, or SubPols. This meant the unwinding

<p>New OAP</p> <ul style="list-style-type: none"> • Organisation Assurance Policy • creates SubPols • any form of "organisation" • in DRAFT for USA and Euro corporates • bar: testng, training, Assurers

of the assumption that organisations are only formed according to local law, which paved the way for the different forms to challenge and be recognised. To do that, OAP says that a SubPol has to be written to recognise each form. These are are mostly on national lines, but do not need to be. Currently, there are works in progress for the USA and Europe.

It also specified some measures of dual-control, and raised the bar on the administration: all involved have to be Assurers, and tested and trained.

More Cracks

Organisation Assurance was unfrozen by this new policy. However, as the audit itself remained frozen, OA was never properly tested. Meanwhile, over time, new information emerged that underscored more issues:

- The process of Organisation Assurance did not get the good documentation efforts that Individual Assurance had, and it remained poorly documented. In particular, there is no Handbook for Organisation Assurance to document the practices.
- Although the OAP raised some new standards, there was never any news that indicated these new standards were being pushed through to the coalface.
- Complaints emerged that the organisations had a special feature, being that they could set the certificate's CommonName to anything they desired. Corporate users will see the sense in that, but Individual Members asked how it was that there was no control on this process?
- Given the success of the risks, liabilities and obligations project, and the consequent Agreements, a new realisation came about: the policy did not make it clear who was responsible for certificates issued to organisations: the organisation itself, the administrators or the holders? Given the normal ability of organisations to duck and weave when the trouble starts, this was no good.
- Another complaint emerged: that organisations had an ability to mass-manufacture certificates. This was felt to be crucial, but it also raised questions as to responsibility for keys. As the Individual Members had strong obligations to look after keys, something was needed on the Organisation side, else imbalances would occur.

Cracks...

- lack of doco
- action on the OAP?
- Orgs can set CommonName to anything?
- who was responsible?
- mass-manufacturing ?

The all-knowing, all-seeing Certificate. Or not?

To resolve at least one of these issues, we examined the question of whether an Organisation could set the CommonName arbitrarily. Org people wanted it so, others did not. After some months, we gradually were able to reduce the discussion to one question:

Sub Rosa

- Is the Name reliable or not?
- Orgs can set the CommonName how they like.
- arguments both ways
- Policy Group: **all info is Verified**
- onus: define how CommonName is verified

Is all information in the certificate Verified or is it not?

This might have gone either way. From a PKI perspective, there is no reason for all the information in the Certificate to be verified, partially or totally, as this is the domain of the CPS and the Relying Party Statement. What is verified is a business decision of the CA; the point of PKI certificates is to deliver claims, and those claims are meant to be documented and defined in the CPS [16]. It is a central tenet of the PKI architecture that relying parties read the documentation and rely on those claims as they are documented. As we saw [above with Names](#), claims can be strong or weak, and this is distinct from their utility.

With some too-ing and fro-ing, this was pushed around the policy group until people realised the simplicity of the question, and voted:

Aye!

Yes, as it happens, in CACert, all information in the certificate is to be verified, and everyone without exception agreed on that. Which solved the CommonName issue *in principle*; now the Org people have to figure out how the CN is verified. And document it.

The Relying Party Statement

As a postscript, this also solved another great need, the Relying Party Statement. Because there was so much emphasis in the criteria on the ability of people to really know what was going on, I felt that a strong, simple and clear statement was needed. Armed with the above, the Relying Party Statement fell out fairly simply:

What do Certs say?

- what can we rely on?
- Relying Party Statement:

**All information in the certificate is Verified.
Certificates are only issued to Members.**

- chain: Assurance ⇔ Certificate ⇔ Reliance

All information in the certificate is Verified. Certificates are only issued to Members.

Followed by pages of notes explaining why these things are important, of course.

Time

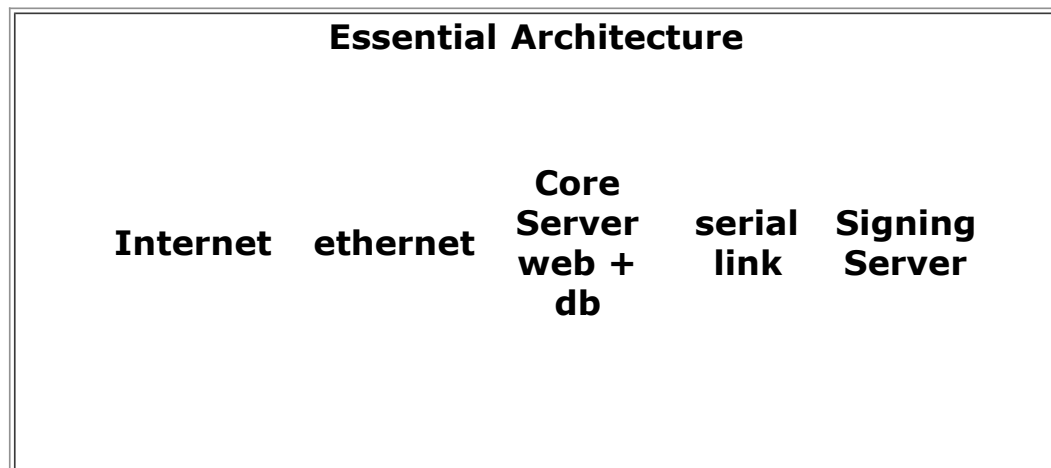
This one issue took around two months to resolve, and the rest remain unresolved. The people involved in OA were not able to resolve these complaints and shortfalls in anything approaching reasonable time, and the policy itself was now seen to be full of holes. For this reason, Organisation Assurance remains out of reach for audit purposes.

Slowing of Audit

- One issue took months.
- Too slow.
- Audit is delayed
- OA will miss the boat.

9. Systems

The systems of CAcert are fairly simple. One small server signs the certificates. One big server runs the PHP-website, and the database, and includes the link to the signing server. This latter link does not use standard internetworking stack protocols, and instead uses a custom serial byte-by-byte protocol.



A Slightly ad-hoc Security Modelling

This then requires strong controls over the systems: physical and logical access control, and security from Internet attacks. Beyond the obvious -- protection from hacking -- this audit has concentrated on balancing out protection against the wider and higher threats. This firstly requires an attention to the threats, then a view over the defences.

The Threats

Threats for CAs are collected approximately into these three buckets:

Threats

- bad certs

- **Bad certificates** resulting from hacking, account compromise, malware, code bugs, social engineering or the like.
- **Data breach** or more precisely the loss, breach or otherwise compromise of lots of user data.
- **Root key**: loss or breach or otherwise compromise of the root key or any subroot key.

- data breaches
- root keys

Against which the CA mounts the following defences.

The First Line of Defence - Community

One thing at its core dramatically helps CAcert to achieve a suitable security level: the persistent focus on community. This created the climate for several defences. Firstly, it paved the way to expressing in the CPS that CAcert served its *Community*, as well as any *community-minded organisation*: It delivers certificates that were good for small businesses, internal security, not-for-profits, and other small organisations. In contrast, CAcert does not recommend its certificates for high value ecommerce.

Secondly, the very strong relationship between all of the Members, backed up by a real forum wherein disputes can be dealt with, meant that reliance is much better defined, controlled and contained.

- 1st Line of Defence - Community**
- certs are "for us"
 - reliance between Members
 - control harm done outside
 - dispute resolution binds it together

Thirdly, that same ability, bolstered by principles, will control harm done to those outside the membership, albeit at the industry standard level of financial liability: zero.

This line of defence covers fairly well the case of bad certificates. In the event, file a dispute. If security of the website or the user accounts or procedures is found faulty, this will come out in the case, which will then identify the area, and feed through a fix; the loop is closed.

The Second Line of Defence - Data reduction

In the event of individual data breach, the Community defences outlined above covers the territory, because it puts in the strong feedback mechanism that finds the best balance.

- 2nd Line: Data Reduction**
- oriented to mass breach
 - data: website, Assurers, certs
 - attention to Identity Fraud
 - *what info?*
 - no photocopies, ID#s, credit info
 - Date of Birth

That line deals less well with a mass breach, in that, if all or most user data is compromised, it is possible that any remedy might overwhelm the Community. Luckily, the organisation has always maintained a philosophy and style as a *privacy organisation*, and has fairly strong controls over the privacy of individual data. This has improved over time:

- Assurance information maintains very limited personal details: name, date of birth, email address.
- This is in several places:
 1. In the database, so systems security and privileged access control is indicated;
 2. in CAcert Assurance Programme (CAP) forms, on paper, distributed across all the Assurers of each person, so any breach of this information is likely localised and physical; and
 3. distributed in each certificate, which information is then presumed to be published [17].
- Careful attention to what information could likely be lucrative and worth of attention. That is, if something could be used to participate in Identity Fraud, we look more closely.
- Hence, in the past, more information was kept, but now less so. The following are (now) not permitted:
 - photocopies of identity documents,
 - identity card numbers,
 - credit card information.
- Copies of (now) not permitted data are to be destroyed.

In today's security world, data breach is probably the number one concern. For this reason, this audit has looked very closely at this issue. The only weak point here, in my opinion, is the Date of Birth, which is frequently and shamefully used as personally identifying information in other online systems, hence has some use in Identity theft. This was strongly debated within the CA, and ultimately the CA concluded that keeping date of birth is an acceptable risk.

The Third Line of Defence - Online Systems Security

CAcert employs the normal approach an array of firewalls, LAMP, SSL, SSH and so forth. The sore thumb is PHP, which is not really a language noted for its security pedigree. This weakness was seen in a recent public disclosure. The good news was that the board responded quickly and well; the bad news was that the bug was fundamentally old and well known: (`register_globals`) This is a black mark.

3rd Line: Online Security

- LAMP + SSL + SSH + firewalls + ...
- PHP code recently broken
- Board response: quick, thoughtful
- Good: proved response existed
- Bruce Schneier: enough real world examples
- work to do on the code?

But there is a more important result: the process is now in place, and is working. Bugs are being found and fixed. This is good. In any system of security, a breach is a positive thing, because it gives confidence that there is a security process in place; whereas the absence of any event eventually perverts any good security into a facade.

The Fourth Line of Defence - Systems Architecture

The problem with all classical anti-hacking systems security approaches is that software is often not as good as one claims, and software is too hard to check easily. For this reason, good software follows the old military principle of *defence in depth*. That is, many layers, many nodes, all of which can be breached individually without causing disaster.

4th defence: Security Architecture

- defence in depth
- signing machine over serial link
- custom protocol
 - TCP/IP stack attacks
 - easy enough to control and review
- root keys

Primarily CAcert's architecture separates its system into two machines, being the online plus database server, and the signing server. The two servers are connected by a serial link with a custom protocol. This is for two reasons: it eliminates internetworking (a.k.a. TCP/IP stack) attacks, and it reduces the protocol elements to those essentially necessary for the job, so it is relatively easy to log and review.

These steps break the problem down into tractable lumps. As above, a network-driven attack could lay waste to the online system, but it could only request certificates over the serial link; these are logged on the signing server, so the damage is firewalled.

The Fifth Line of Defence - Governance

Доверяй, но проверяй

Russian proverb "doveryai, no proveryai", or "Trust, but Verify," popularised by Ronald Reagan.

Finally, we turn to the biggest, most common and most destructive threat of all: insider attacks. This is the province of governance. It is also the major threat against such issues as root key compromise. Measures that are in progress or planned include:

5th: Governance - the inside attack

- Oophaga ↔ CAcert
- root key
 - 2 admins
 - offline root
 - revocation at business level
- vetting of sysadms

- Separation between Oophaga admins (responsible for hardware only) and CAcert sysadmins (responsible for data and services only).

Доверяй, но проверяй (trust, but verify)

- Dual control over access to the signing server (and its root key) at logical and physical levels.
- Separation between primary server team and signing server team.
- Offline root, under dual control, managed by board. Online subroots, so revocation and roll-over of the subroots can be achieved.
- Protocol for revocation of the root discussed and documented with Mozilla.
- Vetting of systems administrators for conflicts of interest.

These and many others are a work-in-progress, evolving both in practice and in documented form in the Security Manual. There remains much to do.

Interactions and the HSM

"Today we were unlucky, but remember we only have to be lucky once. You will have to be lucky always."

Provisional IRA statement to Thatcher govt after Brighton bomb, 1984 [18].

The above defences also interact in curious ways. For example, CAcert uses a separate server for signing, but does not use commercial High Security Modules (HSMs). While promising an improvement in governance through the elimination of the insider-root attack, there are three strikes against: Firstly, HSMs makes a promise of absolute security, and as *"absolutely secure systems do not exist"* this moves the HSM into a sort of fairy-land architecture space that is unreliable [19]. Secondly, the cost is both unknown and very high [20]. Thirdly, the Community line of defence changes the focus of CAcert's attentions, and it makes the fairy-tale of absolute security a story not worth repeating. CAcert does not need to chase the highest "military-grade" security rating, instead it delivers usable, practical and cost-effective security to Members.

The CAcert Systems Story

Scorecard - start 2006		
Community	★ ★	concept no foundation
Data Reduction	★ ★ ★	privacy org, reductions, lack of follow-through
Systems Security	★ ★	presumably in existence opaque
Architecture	★ ★ ★	architectural separation weak review of source
Governance	★	Association + Board lacked control

The Requirement for Control

In retrospect, it is now possible to map where CAcert was at the beginning of 2006, when this audit started. One thing was very clear from the beginning, and that was the general requirement for business control and governance over the systems. That is, not only security from an external hacker was important, but internal control over a rogue systems administrator or some other internal attack was considered paramount.

Indeed, this issue was raised even before I agreed to do the Audit. In early meetings, potential auditors stressed that their approach would be to strongly pursue the core principles of *dual control* and *4 eyes*. Criteria, general industry practice and CAcert itself agreed as well, so it is important to look at why this was so hard to do, and why it took so long. Indeed, if there was one killer issue in the entire audit it was: lack of control over systems.

Governance - Some Classical Techniques

Most problems with systems, and indeed with most businesses, come from inside the organisation. Defences against these issues are frequently termed *governance*, and involve a series of tools and tricks to help the people protect themselves and the organisation.

4 eyes is achieved when Alice watches the actions of Bob. *Dual Control* is achieved when both Alice and Bob are required to act in order to complete the task. Variations exist.

Techniques

- 4 eyes ⇒ dual control
- Escrow
- Logging
- Audit

Escrow is used to keep spare copies of important passwords and backups available to management. This addresses two threats: the potential disappearance of a key person, and the unfortunate incentive that control over systems gives to the rogue worker to make decisions that exceed managerial, business or legal bounds.

Logging records events, or facts about the events for later checking. Hopefully, the records themselves are resistant to fiddling, and the records are checked from time to time.

Audit is used to verify that documentation exists to high-level objectives into procedures, and that the practices follow that documentation. Generally, it is done by a combination of external independent agents and internal, focussed employees.

Separation of Concerns, especially Decisions from Actions. Generally, this permits a broad view on the decision making, and a narrow view on the implementation details.

Although these concepts are well known and universally accepted in professional organisations, implementing them is not easy, and not cheap.

The Plan for Physical Control

At the end of 2006, to add to the woes of unapproved new documentation discussed [above](#), the failure of the original hosting relationship in Australia triggered a crisis within CAcert (discussed [elsewhere](#)). Luckily, by then, there were sufficient additional experienced people involved to help founder Duane, and a plan evolved quite quickly. In hindsight, serendipity played her part, and afforded CAcert the opportunity to get high quality, secure and cheap facilities with a friendly partner in the Netherlands. BIT, a hosting company based in Ede, stood up and offered a rack in their bright shining new and near-empty hosting center, with bandwidth. As well as this, Tunix, a local high-end managed firewall company, and international suppliers Sun and Cisco joined in to fill the rack with the physical bits and bobs.

2006 Crisis

- collapse of Australia hosting
- Dutch plan: ISP, admins, machines
- Oophaga Foundation



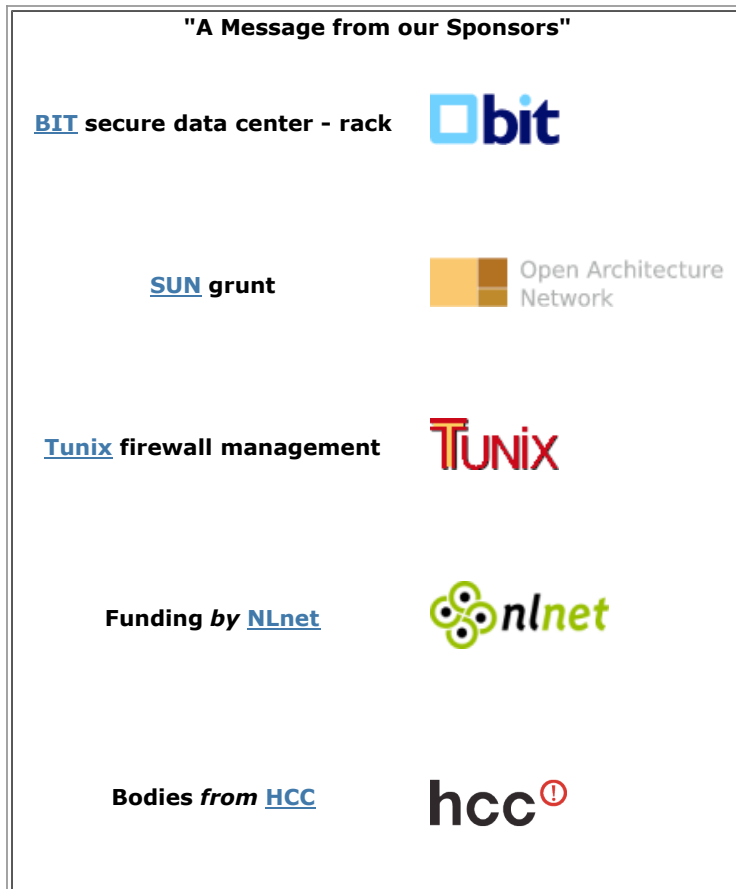
Dual control and 4 eyes were seized on enthusiastically, and "our man

in the Netherlands", a long time Unix and open funding character, rounded up a local group of sysadms from the *Holland Computer Club* to form the *physical access control team*. These people were tasked with control, access and maintenance to the hardware, but *never access to the data*. Their access is controlled by BIT itself. Under a written protocol, the physical team members may accompany sysadms from the logical team in to the data center, and up to the rack to get console access.

This entire team, the agreement with the various suppliers, the protocols and rules for access, and the physical assets under control were then wrapped up into a purpose-created non-profit foundation named Stichting Oophaga [21]. This was bootstrapped from funding by NLnet and XXXX, two foundations in the Netherlands with budgets for good works.

The entire plan came together over a 6 month period from October 2006 to March 2007, or so. It was complete,

comprehensive, and would probably present good account of itself before audit and other similar inspections. It was all good standard and familiar stuff, and suffered from only one tiny problem: the servers were not actually in the Netherlands.



Where Machines Went

In the Christmas period of 2006/2007, tensions were running high due to the impending failure of the Board, and the imminent unravelling of the local hosting facility. A feeling of panic was in the air.

As the above-mentioned Oophaga plan was not coming online quickly enough, CAcert decided to move the machines to temporary locations in Vienna [22]. As this move was looking decidedly impromptu, the episode played a material part in the decision to freeze the audit, until we could at least get the systems back into a state where some sort of review were plausible.

Christmas in Vienna

- Events moved too fast
- Fast decision for Vienna
- Hosting, machines found.
- Fast move over Xmas period
- Good job, but unauditible

Machines were found, a hosting location was acquired, and the systems were moved over the Christmas and New Year period over a few weeks. This seemed like a tough job, done under pressure, and done well.

Getting Control of Machines

Unfortunately, there they remained, in a temporary rack somewhere in Vienna. When the Dutch systems came online a month or so later, the team converged in the Netherlands to attempt a transfer. Then, the issues started.

- **Remote consoles I.** In order for systems administrators in remote locations to manage the machines, they would need some method of controlling the console. Luckily the Sun machines had some form of daughterboard that could control the console, with appropriate software. Unluckily, this was a mish-mash of Java applets client-side display, home grown protocols and the overall result was rather brittle.
- And, unfortunately some home grown security. Just looking through the configuration menu was enough to raise doubts as to the overall suitability for a hard-core security mission, and experience was not any kinder. Later on, CAcert were informed by helpful support people that the product was not to be thought of as aimed at security markets...
- Securing this software then led to the employment of SSH tunneling so as to get some uncorrelated confidence that all was OK. Then, however, the firewalls in employ turned out to be highly oriented towards VPN access and not SSH access. The continual series of manual configuration changes caused tensions and drained energy.
-

Double Dutch

- remote consoles
- console (in)security
- SSH tunnelling
-
- some other problem
- remote KVM switches
- versions, data expansion

Clever! Yes, this section alludes to an issue that is not written up in this report. One day it will be, because openness is the policy.

- **Remote consoles II.** The combined weight of console work led to a search for a KVM switch that worked over the net. It was reasoned this would provide an easy solution because it would disintermediate all of the above issues. One was duly sourced, only to discover that its security was a facade. Much research later revealed that its market was only considered to be within the silo-grade environments; not on the wilder net.
- Version differences between the machines led to fairly serious increases in work-load, having to create strategies of control for the 3 or 4 different machines.
- Meanwhile, the data requirements of the system had also conspired to increase console visits, which climbed to be approximately one per week.

Getting Control of Systems Administration

I love deadlines. I like the whooshing sound they make as they fly by.
Douglas Adams

The above issues led the systems administrator responsible for all this to suggest that the systems should stay in Vienna, and that the enormous work load in just getting access to the consoles in Netherlands would be better re-directed to more useful things.

And there in Vienna they stayed. At this point, somewhat of a divergence developed between the "Vienna" perspective and the "Netherlands" perspective. The former perspective stressed convenience, efficiency of access and the rather compelling advantage of running systems. The latter perspective was keen to exploit the strong physical control, availability of more administrators for governance, and the high profile boost from working with partners.

Divergence

- problems getting control of Dutch machines
- Vienna: efficient access, running code
- Netherlands: control & governance, higher PR
- Board affirmed for Netherlands, September 2007.


At the major meeting in September, 2007, matters came to a head, and the Board was resolved to move the servers by the end of the year. Then, in an October repeat of the Australia story, the "friendly company" hosting in Vienna unravelled, and CAcert was required to secure yet another hosting deal on short notice. The Board seized on this and negotiated a couple of months emergency hosting with Funkfeuer, a community hosting foundation in Vienna, and gave the systems team until the end of the year as the deadline. The Board also took advantage of the physical keycard mechanisms at Funkfeuer, and thus put in place a rudimentary form of dual control in place with



Funkfeuer sysadms providing physical control over access and another local Foundation, Sonance, providing contractual control. It was an advance, on paper, and although it was brittle and unprofessional, it was accepted as good enough for a month or two. Until the systems moved to the Netherlands.

Yet, by the end of the year 2007, only the non-critical services had been moved, leaving the "critical" machines in Vienna. During the ensuing months work was done on the data explosion to reduce it and to better manage it so that frequent console visits were not required. This work was more or less declared complete by around March of 2008.

Austr(al)ia Rhyme

- Collapse of Friendly Hosting
- [Funkfeuer](#) + [Sonance](#).
- Board: Deadline: end of year
- only non-critical systems moved



An Alternative Plan

By this time, the perspectives were becoming more entrenched. Vienna had worked out so far, why not stick with it? At least, that was the reasoning in Vienna. In deliberations on this, the Board discussed and agreed to create a new team with the express purpose of transferring the services to Netherlands. Budget was allocated, a team leader was press-ganged and put on standby for the long-haul flight into the zone.

However, this plan fell over at the first hurdle. When a difference of opinion on some random security issue occurred, the difference bounced up to the Board, and the Board found itself unable to clearly resolve the issue. Without clear support, the plan folded.

Project Cachaça

- new team just for move
- funding from audit work budget
- hit roadblock: security opinions
- physical security declined

Another issue was that while Funkfeuer did a good job as far as its purpose in life was concerned -- a hobbyist or community hosting provider -- it had always been marginal as far as physical security *for a CA* was concerned. While quiet negotiations were begun to consider how to boost up the the raw physical security, with mechanisms such as a purpose-built locked cage, the dual control was failing. Dual access control was always a stop-gap measure, as there are easy back-doors in community systems. Somewhere around May, the front door was opened when a keycard was helpfully given to the CAcert systems administrator.

The Last Chance

Probably, the failure of physical dual control in Vienna was the straw that broke the camel's back. At this stage, the Board took on a more unified approach. Audit also realised that the overall security was deteriorating, and not improving, and this was not part of the deal to give breathing space to CAcert to sort out its short term difficulties. In a series of increasingly aggressive messages, I pushed the situation from uncomfortable to unacceptable to downright crazy. With no alternatives in procrastination left, these statements were made in mid 2008.

Impatience mid 2008

- **Audit Fail** for Vienna
- Roots **Audit Fail** (again)
- new team needed
- deadline: end 2008.

1. The Vienna CA was declared **Audit Fail**.
2. An entirely new team was to be put in place of the Dutch servers.
3. An earlier unpublished advice to the Board was published: that the existing roots were also declared **Audit Fail**.
4. The CA had until the end of the year 2008 to get the systems moved and under dual control, else the entire audit process would terminate.

The Board then took the ultimate and unanimous decision to shut down Vienna servers and move the services, come hell or high water, 30th September. A new team of Dutch systems administrators was recruited, ready to receive.

The decision was taken not to bring up the Vienna servers, *even if the move failed*. The plan called for the Vienna drives to be secured in a remote location under control of a new group of trusted people. The machines were to be unpowered, disassembled and taken out. There was no going back.

This would have effectively meant the end of the CA, and that was fully understood. This finality was clearly written, and it may have signalled just how low confidence was: if the new team failed, CAcert would remain off, and at the AGM to follow a month later, a future CAcert without a CA would be discussed.

This last chance put a lot of pressure on the new team, but more importantly it tipped the balance and got the data moved. In the event, on the 30th September, 3 CAcert people brought the systems down, and packed up the data. One set was escrowed in the safe of a local business; the other set was driven by 2 CAcert people to the Netherlands.

High Noon
<ul style="list-style-type: none">• new sysadm team in NL• end date for Vienna• no fallback• 30th Sept: systems moved to NL!
No Problems!

24 hours later, the data was handed to the new team, and within another 4-6 hours, CAcert was up again.

Why did this happen?

Why did apparently well-educated experienced people simply not follow the basic principles outlined and agreed at the beginning? Why did a community of 100,000 members apparently demand an audit and then decline to help complete it? Why did an organisation with a fair share of senior managers not deliver on its agenda? Why was approximately 18 months wasted while this, that and the other were tried, all while the global clock ticked for the CA, for the community, and for every one closely involved?

Why was 18 months lost?
tough <i>but common</i> questions...
<ul style="list-style-type: none">• did not follow the NL agreement?• managers did not manage?• demand an audit, but ignore it?

These are tough questions, but I feel compelled to attempt a perspective on them. This is partly because CAcert is not that different to any other similar organisation in the open source world. What happened to CAcert is almost certainly relevant to every open source organisation, and I have personally watched the same deadlock in two other big-name organisations in the security field.

The bad news :- (

Volunteers. Everyone is a volunteer, more or less. Although this speaks highly to loyalty, devotion and quality, it also reveals a weakness: nobody can be fired. Hence, if someone decides to go their own way, then they can. The worst that can be done to them is that they are ignored. In the open source world, this is not such an issue, because bad code generally gets discovered, and bad product gets replaced. Even in the business side of a CA, there is less of a problem because a badly written document can be ignored or replaced. These things are even desirable, as good code comes out of independent experiments, and good ideas come out of strange places. But it is a huge problem if they are operating the root keys.

Technophilia. There is an almost complete focus on all issues as technical problems with technical

The bad news :- (

solutions. When non-technical problems present themselves, attacking them with technical solutions is risky: This occasionally works, and sometimes innovative solutions appear; indeed, it works just enough times to reinforce a belief that all problems are technical. But it fails as a strategy for problems for which there is no technical solution, or for which we haven't yet invented the tech approach, and there are a lot of business problems in this class. There is a reason why MBAs do not learn rpm(1), C, or crypto, and absence of knowledge of these things should not be seen as an excuse to ignore them and download yet another package or code up yet another feature.

- Volunteers
- Technophilia
- Human++
- More is Better
- Business Coding
- Distribution
- Founder's Paradox

By way of example, dual control, by its very nature, requires two humans, and they must be dealt with in a human fashion. A focus on tech only serves to distract from this essence. In fact, all of governance is essentially a set of issues requiring human solutions to problems which we cannot yet solve in a technological fashion. Technology solutions applied to governance problems seem to have a very good record of compounding the issue unless they are very well thought out, or lucky.

Human++ Technically-focussed people tend to be weak at that which is called Human Resources. Especially, those who have earned their spurs in Internet times, and have worked in open source world, tend to see any such discussions as akin to voodoo or tarot cards. This tends to lead to an alternate ceremony, where small teams of expert people discover they need more people, agree they need more people, and do anything but engage new people. Generally, the ceremony repeats with more work, more agreement, more invoking of tarot cards and voodoo, and eventually burnout and sacrifice of key players. Humans do not currently download in packages, although I am sure there is bug filed about this.

More is Better. Fourthly, there is a presumption in the minds of many people that all are trying to do the best job possible, which is probably true, and that this means that whatever people do is probably good, which is as probably false. If anything, the Internet is a massive cauldron of wasted efforts, out of which only the best survive. What happened to the rest? The bounty we enjoy from the Internet process is wonderful, but it should not distract us from the observation that much of the work is wasted, inefficient, wrongly directed or downright dangerous. Why this should be mysterious is beyond me; most code that is written is much the same: inefficient, wrongly directed or downright dangerous and hopefully wasted. I include my own.

Business Coding. There is a hubris about code that inspires the junior techie to launch into convincing and confident displays of managerial talent, because the code works and the people use it. Unfortunately, there is a slight yawning gulf between *running code* and *running business* that is not really covered in the install manual from that most recent download. The absence of instructions on how to fly is taken broadly; leaping casually into law, interfaces, psychology, finance and other mysterious and beautiful gulfs is common in the technical world.

The closer a person gets to the code -- coredumps! bugs! patches! SVN! -- the more a developer is consulted, and expects to be consulted, on all issues under the sun. The further someone is from the code, the more their voice is pushed away, "he doesn't code, he's no help." In the corporate world, this hubris is more benign as the techie is not asked to launch into new markets, or repair existing ones; but the open source world has yet to pick up on the full market cycle.

Distribution. CAcert, in common with a lot of open source organisations, is distributed widely. The core players derive from 6 countries and 3 continents. This has the advantage of being able to tap

skills available from all over the world, but it has the disadvantage of reducing communication. Without the face-to-face, and without the persistent presence of the manager, issues tend to roll on without resolution, ad infinitum. It is hard to see how to deal with this without a fairly big budget to move these volunteers into a single locale. An approach that can help is to give local teams one entire area. For example, systems administration is now concentrating in the Netherlands, and the education campus was a German project. However, it is important for each regional team to see themselves as players in a global world, not as exporters of their own culture.

Founder's Paradox. Finally, CAcert is the product of the entrepreneurial process. Innovation caused it to happen, and the innovator that started CAcert was essential to make that leap into the unknown. Yet, the same innovative spirit that makes the jump against conventional, stagnant logic possible in the first place also blocks the movement across to professionalism. This is called the Founder's Paradox, and is no different in the fully commercial world. Many a great company has been laid low by this failure. In a way, CAcert was lucky: earlier, wiser voices forced the Association on the Founder, the audit process broke the deadlock, and Advisory picked up the pieces. But the cost of this is too high to take cheer in it.

Is there a One-Liner?

It is all too easy to say: the systems people did not know what they did not know, but they were certainly inspired in their belief in it! In time, the overwork and the hubris conspired to shut out conflicting opinions, and build a castle. Students of Boyd will recognise the self-reinforcing loop syndrome, whereby confirming data is amplified and disconfirming data is ignored or attacked.

The Board of CAcert had to learn this. They had to get over the honeymoon period and find out for themselves that the work being done on the systems side was flawed by a too-insular, introspective view. Although it was "obvious" to some, and CAcert saw experienced managers and techies come and go because of this very frustration, only when it became a shared truth to all was action possible.

But! The good news!

This is a bad story. It cost the audit, myself, CAcert and every member around 18 months of delay. To temper this, I can offer some comments on the positive side for CAcert.

Firstly, CAcert has faced this issue and dealt with it. It now has a management team and structure that can deal with such big issues, and it is now moving on. CAcert is certainly not up to the level of professionalism that is standard in big companies, but it does know what is required, it is fighting to get there, and it has a fighting chance of actually making the grade. It has a shot at the title, which may be a first.

Plus Points

- This Mountain has been Climbed!
- It's lonely up here: no other OS orgs
- The certificate mountain!
 - risks, liabilities, obligations
 - reliance + utility
 -
- Lonely up here, too!
- Nice maps, but ... they should be!

Secondly, in my observation and opinion as a professional manager, no other similar organisation has bettered this. Indeed, where evidence presents itself to me (two recent cases that support direct comparison) it tends to confirm that the other big open organisations are not yet even at the point of knowing that this gulf is an issue. Lack of management and the consequent failure to deal with anything outside the pure technical domain is the rule, not the exception. With practically all large open source groups, they are experts in fixing bugs, quoting RFCs and delivering software, but ask them to discuss issues

that effect people, not lines of code, and they flounder. Indeed, one observer who has the inside track on many open source organisations recently commented privately to me that *no open source organisation had yet made the jump to a professional management*. To be fair, many open Internet organisations do not need it, because they are not under the imposition of an external audit or other controls, and they are not delivering a product that is integral to the user's security. But, some do have "security" in their mission, and they need something better.

Thirdly, this may bring cheer to those critics who say that CAs should be professional and commercial organisations, and an open community has no business entering into this security area. The response to that is clearly, yes in part: the ability to hire experienced managers, fire inexperienced ones, and the overall feedback mechanism of the competitive market give those players an advantage. People and business problems will be solved, or the company dies.

Yet, no commercial CA has solved the challenge of delivering a useful end-to-end claim that gives some form of tangible and decided benefit for the users of certificates. Whatever they deliver may be very efficiently and professionally issued, it just lacks any ability to impress the end-user.

Finally, it should be pointed out that the only reason the business side of CAcert looks so good and professional is that it was given around 3 years to develop the structure, concepts and documentation in relative peace and quiet. If the systems had been ready in 2006, then the lack of documentation would have killed the audit. Stone cold dead.

Wheretofore the systems?

What now? Sometimes when climbing a mountain, we forget that we have to climb down again, and move to the next one. Let's check the map, as of today, November 2008.

Physical. The movement of the systems to Netherlands, now completed, creates a regime of *physical and logical separation*. Oophaga is responsible for controlling the physical access to the hardware, and will be responsible for maintaining the governance controls over that physical access. Once complete, this process should be good, and should withstand an audit scrutiny.

Current Status

- Physical: ready for check
- Logical: close, possible.
- Doco: CPS + SM lack approval
- X-team: future task to develop

Logical. The leader of the sysadms is responsible for placing the logical access to the critical systems into a governance regime. In the past, this was impossible because the team was just one person. Now the team has more people, so governance is possible. New rules are now in place to dual-control the logical access over the signing server, and 4-eyes over the primary server. There is a limit to how fast we can expect the new team to catch up, but once they are ready, we can begin. (If you are in the Netherlands, and have sysadm skills, are not involved in any conflicting work, then consider giving a hand.)

Critical Documentation. The audit identifies and requires three critical documents that are currently all well advanced, but not quite there yet:

- **Security Manual.** This should be the documentation that will bind the different areas together in the security role;
- **Configuration-Control Specification (CCS).** The same requirements that are envisaged for policy documents such as the CPS are also imposed on software and hardware. Traditionally, control processes have treated these different classes of assets in very different forms. It remains to be seen how the CA will deal with this "simplification" but there are no obvious obstacles. It is simply a matter of documenting what is done now, comparing it against audit and other criteria, and improving it.
- **Certification Practice Statement.** The CPS is the traditional primary document for audits and communicating with the users. This is well advanced.

Security Oversight. The security manual crosses across the entire CA,

and thus requires a similar broad focus on the part of the people. In the future, as the new sysadm team and SM beds in, there may be a requirement to develop the skills in a cross-CA basis. This is clearly beyond the scope of the systems administrators. The CA needs to develop its broader security management. Because a CA is a security organisation, and because all aspects bear on security, expertise in this area is an imperative.

Scorecard - end 2008		
Community	★ ★ ★ ★	promulgation!
Data Reduction	★ ★ ★ ★	DoB
Systems Security	★ ★	1 public bug, more?
Architecture	★ ★ ★	future target
Governance	★ ★ ★	New team work-thru

10. Audit

What went Right

The Audit, following the lead of David Ross's criteria (DRC), correctly identified an archilles heel of certificate issuance: risks, liabilities and obligations. Even better, this challenge has been met with a strong Community allocation of liabilities, backed up by dispute resolution. Adding those to the CAcert Assurance Programme, itself improved by policies, led to a thing never before seen: a clear story on reliance.

Good Stuff
<ul style="list-style-type: none"> • Risks, Liabilities, Obligations ⇒ <ul style="list-style-type: none"> ◦ allocation by Community ◦ Assurance by policy ◦ dispute resolution • Arbitration for all the edge cases • Education Campus • Management: Board ⇔ Policy ⇔ Arb

Arbitration is a big achievement. Following on from developments in other communities, this provides a closing of the loop that simplifies a lot of CAcert's operations.

The management aspects were heavily influenced by the demands of Audit, and although painful at the time, I think it undeniable that the result is far better and stronger than the past. CAcert's triumvirate of governance tools: the Board, the Policy Group and Arbitration work together like a three legged stool. None is too powerful, they all work better together.

The work on the Education Campus to create a testing regime is another highlight. Although basic to start off with, testing and education promises a lot for CAcert in the future.

These innovations all can proud of.

What was not Covered

There are a few areas that were not strongly covered.

Yes, another tricky black bit. One area was not covered deeply, and as this is a complex and dynamic area, this review does not comment for the moment. Better to let it finish.

Finance. This audit did not look at the finances of CAcert.

Board and business. This audit was not supposed to look at business aspect. However, more aspects than I would like were looked at.

What went Wrong -- I -- Openness

This was supposed to be an *open audit*. I always wanted to do an open audit, I'm that sort of guy. From my experiences in payment systems it came out clear and strong: if you want security and governance, the open, transparent possibilities are much stronger than anything that can be done in private. Just ask anyone in Wall Street, they will definitely see the merit of transparency in risks.

This resonates with the open source community, but it goes much further than that. We call it *open governance*, and it means opening up all of the processes, all of the checks, all of the information. OK, not absolutely all. Not the keys, passwords, not user's PII, but ... *everything else*.

What went wrong -- Openness

- DRC unpublished
- work first
- secrecy!

But, this audit failed to open on starting. I have to admit that and count the cost.

DRC was unpublished

DRC was not initially released into the world. Although CAcert had the pre-approval from Mozilla that it needed to start the audit to this criteria, the author had not formally published it. In retrospect this put us in a bad position; we were not able to clearly communicate what is needed. Further, the lack of exposure reduced the credibility of the process, giving the audit a sort of fairy tale remoteness. Finally, it makes us look as bad as those competitors who also keep parts of their audit process confidential, in order to increase their power of negotiation.

Negotiations with David Ross gave us an authoritative copy on his website.

Get Some Work Done First

There was a tremendous expectation on the audit. Everyone wanted one, and everyone believed it was something you could purchase at the supermarket, if you had the cash.

In order to manage those unrealistic expectations, I chose not to publish much detail of the audit at the beginning. I rationalised that it would be better to plough through the first phase, get the documents sorted out, in place, and then announce a sort of mid-stream progress.

Big mistake! It proved impossible to get the documents sorted out. Gradually, I was forced to reveal more and more information in order to spark some activity, by means of wiki and posts. By the time this was done, the chance of some momentum of a big push was lost.

It is not clear how being tight with information at the beginning effected the end results. It certainly did not help, and probably caused a minimum of harm by slowing the process down to a crawl. Information was not channelled and shared, and there was not a lot of help available for the project. So much so that when I was chatting to one of the Board members around November of 2006, he admitted to me that did not even know there was an audit in progress. He also admitted to being severely embarrassed by this lack of information, and it was yet another signal that things were out of control.

Climate of Secrecy

Like too many organisations, the early CAcert was obsessed with secrecy. The management discussions, such as they were,

Obsession with Secrecy

- hard to get anything done
- your attacker laughs!
- ideal cover for incompetence

were secret. Policies, such as they were, were decided in secret. Location of hardware was a secret, seeing the software required an NDA.

Sometimes it seemed that secrecy was the beginning, middle and end of conversations.

- ideal cover for incompetence
- check out the OSS Simple Sabotage manual...
- there are no good reasons for secrecy, just less bad ones...
- CAcert formally adopted a "no secrecy" rule

There are three issues with this. Firstly, it is shockingly hard to get anything done. This audit stalled and sputtered for a year before I had enough evidence to pass sentence on the Board that wasn't. I would have known, and we all would have known, within weeks if decisions had been published. But instead, we had to send pings into the Board with TTLs stretching into the months. We can't fix what we don't know.

Secondly, your attacker laughs at you. The same secrecy that keeps idiots like me from figuring out what is going on also provides cover for the activities of a smart attacker. Secrecy is no barrier to the enemy, and indeed I spotted 2 governance attacks during this period, employing CAcert's own secrecy against it. And these were just commercial-grade goons, heaven knows how many junior spooks the agencies have sent on training runs.

Thirdly, secrecy is the ideal cover for incompetence. Secrecy rarely helps a good plan, but it is an excellent tool to cover up a bad plan. This is pretty much a constant, in that every time I've come up against an issue, and been told "that's secret, you can't see that," I've discovered later that the real reason was that the work simply wasn't correct, wasn't complete, wasn't done, or was something else entirely different.

Indeed, I've yet to discover a good reason for secrecy. OK, secret keys, passwords, user data aside, but, everywhere else? No rationale as yet has surfaced. CAcert was not the first to prove the rule, and won't be the last.

Luckily I was not alone in this viewpoint. In September of 2007, the Board agreed to make everything open and transparent, by policy and by practice. Anything that needs to be secret now has to be justified. Slowly, CAcert has shed its secretive ways and now publishes:

- all its formal decisions,
- Arbitration rulings,
- threat security modelling.

And, there is more to follow, as CAcert learns how shedding secrecy will help it.

"Trust me"

Highly allied to secrecy was the temptation to resort to personal *appeals to authority*, more simply known as "Trust me!" When people are asked about some area that needs to be documented, and isn't, one frequent and automatic response is to point out that the commentator is the expert, and the work is done well.

Trust Me!

- is an *appeal to authority* ploy
- implies we don't need to document
- Audit: either it is written down
- or it doesn't exist!

And, by implication, we are to suppose, we do not need to document it, or discuss it further. The Audit has a different perspective:

Either,

it is written down,

or

it doesn't exist.

Pick one! The polite way of explaining this is to point out that the one person doing the job might get hit by a bus tomorrow, and the organisation needs some documentation to deal with the change. However, the reality is far sadder. In almost all cases, the work is done incompletely, badly, with no quality control, arbitrarily, and with little chance that anyone else will be able to follow it, let alone with any chance of external approval. It cannot be documented without revealing these delitos, so it is important to avoid conversation about what is being done, at any cost.

The cost of this approach is simple: no audit.

What Went Wrong -- II -- Independence

Independence is a central tenet of audits. However, it turns out that independence is at best an impossible goal, and at worst a charade. Where any given audit falls in this unhappy spectrum depends on many things.

In this, independence is just like security; there are no absolutes. We need to understand it in depth before relying on it, and without that understanding, *it will be unreliable*. Here is an attempt to shed some light on some of the challenges to independence, in an effort to better understand the risks that you the relying party takes.

If Crypto were a Physical Science, CAs would be in the Quantum Uncertainty layer

There is no such thing as absolute independence; as soon as the Auditor walks in, he effects the situation. From physics we know it is impossible to measure something without changing it, and auditing displays the "Observer Problem" at a level that is approximately quantum.

What went wrong -- Independence

- "observer problem"
- producer
- product
- criteria
- governance

Sometimes this is for the good, in that standards are lifted and more thought is taken before acting. Other times it is not for the good. People can become unnecessarily bureaucratic, difficult, or scared. In all cases, it makes the real job of verifying much harder to do.

The Auditor is not the Producer

In theory, all the auditor should do is deliver an independent opinion. The organisation has to do the actual work to deserve that opinion. So, if a document is missing, the organisation has to write it, or find it elsewhere. If a feature is missing, the CA has to add it. But shaking the belief that the auditor provides the audit proved next to impossible. From top to bottom, this was the view held. Here are some of the motives why this was so hard to do:

- Only the auditor knows what a document needs to say in order to pass the audit. Ergo, he should write it.
- Not so many people are capable of writing serious documents.
- Almost all of the active people in CAcert were traditionally non-native english writers.

The first two are fallacious. The first fails as if the auditor writes it, nobody else need follow it. The second fails for the same motive; if you cannot write the policy, it also isn't your policy, and in effect, you have no policy.

The final one is tough; as CAcert is a volunteer organisation, it has to deal with the volunteers that it has, and it is only in the last year or so that alternative volunteers who can write in high-grade English have turned up. But it still fails to pass muster as an excuse; as other languages can be used, good language tools these days, and good

english can be turned relatively easily into bad english. Indeed CAcert claims to be a leader in translations!

Either way, this is a grave disappointment. Although CAcert musters 100,000 members on the books, 10,000 Assurer Candidates, and 1000 tested Assurers, it can only find around 10 people who can help with documents and to do work required to help the audit, and far less than that to add the features needed.

The Audit is not a Product

Perhaps the second most disappointing thing (after the above more practical point) was the attitude that the auditor is a hired gun, and he'd better start slingin' for his supper. This took its form in a persistent and deep belief that the audit is a product, and not a process. So the continual question pressed by the entire CA was more or less reduced to the following:

"Has the audit been done yet?"

No such. Many many people believe that CAcert only has value if put in the browsers; and most also understood the convention that an audit is required to get it there. Yet, there were very few people in the organisation that stood up and said "OK, what do we have to do?" It is as if the standard mindset was that the audit was someone else's responsibility, when in fact it is everyone's responsibility, and if nobody tackles it, the game does not start.

One supposes that CAcert is not unique in this. Indeed, if we look at other links in the food chain, everyone outside the process also thinks the audit is a product; vendors, users, developers, CAs, everyone: once you've got your audit, you are good, and without, you're no good. The obsessive and wrong belief in audit-as-product is only matched by its perverted inability to align with the needs of its original market.

The Criteria are Your Target, not the Auditor's

The audit works to criteria, yet nobody was particularly interested in working through that document. No-one ever stood up and said "Where is the list of things that we need to deliver?" DRC is very well written, and is not hard to understand. There are relatively few misunderstandings. Boring, yes, but we need only take a criteria a day if it is too heavy.

Hence it is a surprise that only very few people, to my knowledge, have read through even some portions of the criteria and related this into their areas.

Governance is done by the Auditor, Right?

Governance is something done by the organisation, but frequently there is a temptation to slot in the Auditor in some governance role and imagine that it is done. The problem with this is that (a) it is then impossible to audit that role, and (b) it is impossible to rely on the role being done afterwards. Hence, an automatic audit fail.

This rather human temptation to solve a boring problem with the nearest spare and idle Auditor is not unique to CAcert. Indeed, the EV Guidelines includes this rather impressive blunder:

(e) Root Key Generation For CA root keys generated after the release of these Guidelines, the CA's Qualified Auditor SHOULD witness the root key generation ceremony in order to observe the process and the controls over the integrity and confidentiality of the CA root keys produced. The Qualified Auditor MUST then issue a report opining that the CA, during its root key and certificate generation process:
EV Guidelines, V1.0

If the Auditor is now part of the governance operation, then the ability

to audit is lost. Just the appearance of that statement above will cause everyone to kick back and let the Auditor do all the heavy lifting.

"The big cheese from KP-Anderson-Goldman-Lehman was there, of course it was good! We did everything he said!"

When the organisation cannot be trusted to create its own root with its own procedures and governance, what does that say? Will EV++ say that the Auditor must guard the root himself, and only let the CA see it on Sundays?

Time! Value! Recompense!

This audit took too long. (*It's still going on!*)
An audit should be an economically efficient check, it should not be an open-ended nightmare for all concerned.

Financial (In)dependence

- too long, too much time
- small retainer + expenses
- Money effects independence

This audit has taken three years of my time, and will take more. Although it has not been fulltime, the commitment has still caused issues. There are significant difficulties in doing such a long job without recompense. As there are significant conflicts of interest between any audit and any other activity, there are also severe issues with finding other employment / activities to provide sustainability.

To address this, in the third year, a small retainer and expenses was agreed, thanks to good-works-funder NLnet [23]. This level of funding does not do more than recognise the availability.

Money!

However, once money enters the equation, it immediately effects independence. To recognise this, Mozilla stipulated rule 10 in their "criteria" or policy [24]:

10. By "independent party" we mean a person or other entity who is not affiliated with the CA as an employee or director and for whom at least one of the following statements is true:

- *the party is not financially compensated by the CA;*
- ***the nature and amount of the party's financial compensation by the CA is publicly disclosed;*** or
- *the party is bound by law, government regulation, and/or a professional code of ethics to render an honest and objective judgement regarding the CA.*

To address yet another conflict of interest, I was probably the one who suggested the second bullet point (my emphasis), well before this audit started! No matter good intentions here or there or anywhere else, I can say that money dramatically changes any notions of independence, and if I had any small influence over this Mozo policy again, I would go for this:

10. By "independent party" we mean a person or other entity who is not affiliated with the CA as an employee or director and for whom all of the following statements are true:

- *the nature and amount of the party's financial compensation by the CA is publicly disclosed; and*
- *the party maintains an independent state of mind; and*
- *the ultimate beneficiary of the audit is the end-user of Mozilla products.*

10.b By "independent state of mind" we mean one of the following statements is true:

- *the party is bound by law, government regulation, and/or a professional code of ethics to render an honest and objective judgement, or*
- *the party can show by open documentation and methods of governance that an objective and verifiable judgement is rendered.*

That is, there is reason to believe that if I find it so difficult, it is likely to be difficult in other circumstances as well.

Maintaining an Independent State of Mind

The USA auditing practice recognises the Observer Problem, and states words to effect that independence is a matter of *preserving an independent state of mind* [25].

State of Mind

- approved ⇒ owned; avoid unreviewed
- Who is the audit for? The **end-User!**
- Avoid: other CAs, instructions, tasks

Without that gem of wisdom, it is hard to see how sanity could prevail. All of the above issues threatened the audit process in one way or another, at one time or another. And, they often challenged the independence of the process. I dealt with these challenges to Independence in various ways:

- All policies were approved. Initially, by the Board, and later by the policy group. Hence, even though inspiration was often found from the Auditor, a tough approval process forced the ownership of the results. Practically, few suggestions of the Auditor escaped a fight, and many were bloodied. Often enough, Audit did not get a desired result and was overruled by the consensus that choose a different path.
- Distancing from areas that had no separate review. Some areas were not essentially subject to any oversight or review, especially the systems. It proved impossible to get close enough to suggest solutions without fundamentally impairing independence. That is, the more "suggestions" were made, the more the auditor was cast in the role of manager not reviewer. When faced with the unacceptability of the systems, we had to wait for a long time for others to discover that and for consensus to build up.
- Separating *role* from *person*. As seen here, we used the term "Audit" to de-personalise the issue from myself, and to focus attention on the criteria, the desires of Mozilla and users. We have also used the same process to raise the profile of the Arbitrator.
- Forcing extra people to observe, when the viewpoint of others was that auditor could fill in governance holes. This was especially prevalent in systems areas.
- Identifying who the audit is for. No, really! Surprisingly, this step is rarely done, and hence, I had to create it. In this case, this Audit is for the end-users of Mozilla's software, and to a lesser extent Mozilla itself, but only to the extent that it stands in as a responsible agent for the users (itself a cause for concern).
- Vigourously rejecting every other CA as a model. This was partly necessary to stop natural temptation to just "borrow" the documents and processes of others. In essence, I had to force every decision and policy to be made on first principles, and hence the result looks completely different to the classical commercial CA [26].
- Issuing as few instructions, or directives, as possible [27]. In practice, 4 or so have been issued, and that is 4 too many. Every issued instruction turns up the dial on apparent authority, and reduces the dial on apparent independence.

What Went Wrong -- III -- Audit Process

There are detailed issues that are shortfalls or failures *with the audit process*. These are things for which CAcert cannot be criticised for, but instead they have to suffer, if not in total silence. Here are some of them:

CA Audits are not for the Benefit of the User

For several reasons, the User is not part of the equation:

- The audit delivers benefit primarily to the one who pays. In this case it is the CA, and not the users.
- As described in this document, audit criteria currently fluff the question of how the relying party is to rely, and how the end-user is to be protected. DRC goes some way by insisting that these things be documented, and in the open, but it also stops short of actually requiring any benefit or balance for these parties.
- The statement of user as customer of the audit is not made, or where it is made, it is generally fudged. It is entirely up to the circumstances whether the end-user is considered at all.
- Representatives of users do not typically sit on the relevant committees, appear on developer groups, participate in anti-phishing venues or the like. By far the greater proportion are representatives of companies selling some security product or other. The standard of representation is generally lip-service as far as the user is concerned.

Wrong III - Process

- Benefit to User?
- The map
 - Vendor, Standards
 - Business Risks

The result of this is, predictably, the user is forgotten in the equation. She may be worse off than without the entire process: She is given zero liability, and zero remedy. She is forced to use the certificates, or *not get any "protection" at all*. A subscriber is required to pay for this

"benefit," yet has zero expectation that the players will deliver benefits against the threats to him or his customers.

Audit does not cover the map

The audit is purposed, more or less, to permit the vendor to make a judgement call on behalf of users and itself. This is fine in principle, but there are flaws. Firstly, it does not reach far enough. Audit practice today looks like an 18th century map of Africa, with large bland expanses marked simply *"unexplored."*

Let's explore. On the one hand, the vendor demands an audit, CPS, security measures, etc.; on the other hand, the vendor can present the business model of the CA, or can hide all this from the user. With a few lines of code, the vendor can either work for the user, or turn off the CA's security model. Completely.

Decisions taken over the last few years by vendors confirm this power. Not only does the process put the CA to great cost, it denies the CA any power to influence the security delivered to the end-user, even when it is explicitly written into the CA's model (and follows agreed PKI practices). The vendor therefore exerts a greater influence over the end result than the CA can do, (and, by extension, the auditor). We might argue over whether the vendor knows more about the CA business than the CA, or whether the CA knows more about the users' needs than the vendor. However, one thing is a fact:

There is no audit of the vendor.

If indeed we accept that the audit is a necessary part of the process, this is a considerably vexing question. If the users demand an audit over the CA, then they should demand one over the vendor. And, the two audits -- CA and vendor -- should be aligned, from end to end.

Audit Omits the Security Leaders

"A camel is a horse designed by committee."
Variously, Sir Alec Issigonis, or Vogue

It gets worse: Consider that we have confidence over the vendors, who write lots of code for security motives. Above, we intimated how decisions were taken recently that indicated their willingness to change the game entirely for the CAs. A further issue lies in that the vendors, who hold the power of the code, did not take those decisions themselves, but instead routinely outsource them other organisations.

In effect, security leadership is outsourced by vendors. For example, PKIX does it for the CA interfaces, the TLS committee for the cert and crypto layers and now, the emerging CAB Forum does it for the positive side of the user interface, such as the EV "green" label. The browsers are still preserving the decision making with negative user interface decisions, but this only serves to highlight how emasculated their possibilities are. Following on from the above, we can now note the observation:

There is no audit over the standards bodies.

There is a long distance between the approach of an Internet standards committee and the security of an end-user dealing with a phishing email. It is an open question whether such a body would survive an audit; a recent observation was that audit calls for disaster recovery, whereas the PKIX struggles with a policy of a root as a single point of failure. One is a very standard business requirement, the other is a theoretician's elegant structure. Such elegance would not survive an audit process, and it is only the separation into "in audit" and "out of audit" that keeps the flaw in place. This reflect poorly on the audit result; as the audit is clearly undermined in its work when the answer is "go and talk to committee if you don't like it."

Audit Omits the Wider Risks, Liabilities, Obligations

This leaves the customer of the audit rather exposed, and highlights another weakness that an audit for business risks should have dealt with: Security includes proper relationships between the parties. The Audit makes a judgement call that incorporates the effect of the following entities on the user's security:

- vendors such as Mozilla and Microsoft,
- software committees such as IETF,
- other parties such as CAB forum,
- anti-phishing services,
- and of course, CA and Auditor

The judgement call has wider implications because it relies on liability being accepted between the parties; as there is no contractual relationship with many of these organisations, we must be concerned about the clarity of the liability allocation.

This means that any situation that might involve significant risks and liability being passed outside the CA is going to be difficult to be clear about. Uncertainty leads to costs in any dispute. For example, CAcert's liability position, in common with most other CAs, passes the liability for various actions to the user. Is the user defined? Is the user an end-user, or is the user a relying party? Is the vendor standing in as the relying party (by dint of control of information) or is the browser using a back-to-back agreement to pass on the CA's liability position over to the end-user? Does the end-user know any of this?

There are some answers to these questions, but they are not clear, nor unified, nor transparent. The situation is therefore legally fraught, and the Audit's position on making a judgement call over the liabilities outside the CA's domain is compromised.

Conclusions on the General Audit Process

Given all the above, what can we conclude?

Audit cannot respond to any developing threats. Indeed, this is what has happened. Phishing erupted in the early 1990s, and audit criteria and audits themselves were powerless to respond. DRC does no more than document the R/L/O, and EV does no more than document the old regime.

Wrong III - Conclusions

- new Threats
- model does not change
- improving security is beyond CA
- Costly - against what benefit?
- Other models work!

Audit can only address security within. Because so many of the practices are outside the control of the CA, and indeed outside the vendor's control, there is little that Audit can do to resolve the serious issues for the user security.

Benefit to Users cannot be identified. Because there is nothing much that ties the Audit to the benefit of the user, and because the vast majority of audits are conducted in private, or to complex criteria, it is impractical for any user to determine what an Audit says to that User. As the CA pays for the Audit and the CA has a financial interest in the result, it takes no great stretch of the imagination to conclude that the user is lost in the process.

Audit Costs too much. The costs of Audits are generally hidden, but various estimates have placed it at around \$250,000. CAcert's audit has also cost that much, if you calculate the entire package as including the opportunity cost of the people's efforts.

Other models work. It can't have escaped the notice of any audience that models such as Skype and SSH provide real security to millions of users, without any necessary audit. I am not going to argue in this document about the *mores* or the *lesses*, and it is recognised that

supporters of the audit and PKI model will contradict this claim. These discussions are well rehearsed elsewhere.

The point however is that the audit profession has set itself up in a privileged and safe position, but has not delivered anything in return. This might have been fine except for the evolution of phishing; the challenge that the auditors will have is how to get the users to shed a tear when the audit process finally is asked to show how it is delivering value -- to users. Or be bypassed.

Conclusion

Status of CAcert's Audit

Where are we?

- CAcert has enough documentation to be analysed under DRC-A and DRC-B, for Assured Members. This can be started as early as December, and should be quick, once started.
- Evaluation of Assurers in action will take longer; but as this is an ongoing task, and the process is subject to continual quality improvement, this does not need to delay any other areas. As long as it moves forward, we can carry on.
- New Root and new Individual Assurance subroot have to be created. December.
- However a number of critical acts remain undone which will block:
 - Notification to all Members that they are not Members, under a new CAcert Community Agreement. An option to this is to close all un-notified accounts.
 - Addition of positive "I agree" checkboxes.
 - Turn-off of all the old Assurers.

<p style="text-align: center;">Status - Clear</p> <ul style="list-style-type: none">• Start doc review• Assurance can be checked in parallel• New roots - December?
--

These are required to place the agreement into effect; without them CAcert lacks an agreement, is not a community, and all we have said so far has no foundation.

- Resolution of the difficulties with email/domain probing, then completion of CPS into DRAFT. Probably one month after the previous point.
- Security Manual has to be completed. November-December.
- Systems have to be reviewed against SM and CPS. That will likely take some weeks or months.

<p style="text-align: center;">Status - Blocking</p> <ul style="list-style-type: none">◦ Notifications, checkboxes, old Assurers◦ email/domain: CPS◦ Security Manual work thru◦ Operational review: Feb/Mar

If all the above work out, we could see an **Audit Pass** for the new root and subroot for Individual Assurance within months.

Organisation Assurance has too many problems, so it will slide until the team sort out a new document and resolve the bugs. Unnamed certificates and non-assured certificates can happen when the issues with email/domain checking are resolved.

Recommendations for All

What should Users do?

Because of all the foregoing, it is hard to recommend that users demand or rely on audits. Neither over the vendors, nor the CAs, are audits likely to tell them anything.

<p style="text-align: center;">Recommendations: Users</p> <ul style="list-style-type: none">• Users should not ask for an audit• Instead, ask for offer + name• (CAs should also ask for name.)
--

Instead, it suffices to do this: demand a clear offering from the CA, and demand to see the CA's name. If each user can clearly seek remedy or satisfaction for any action, then we can use the traditional processes of brands (beforehand) and the courts (afterhand) to improve the quality and provide the needed security for a reasonable price.

What should CAs do?

Listen to the above demand from users.

For the most part, this is all in place, so CAs have little to do. Most CAs have these agreements in place, and have had them for years, because

their own legal analysis has said they are needed (and not the audits). Indeed, some CAs have spent millions of dollars on the legal work; we should use this. It simply needs a change in the thought processes of the participants to shift from thinking about audits to thinking about bilateral agreements.

What should Vendors do?

It is pointless to opine on the whether audits should be demanded by vendors, because they will anyway, regardless of the logic or value. Instead, I recommend that the vendors

Recommendations: Vendors

- Vendors: direct audit to *their* end-users
- R/L/O: are vendor's liabilities clear?
- Auditors -- how to add value to Users not payers?

- a. direct the audit process to the benefit of the end-user [28],
- b. write their own audit criteria [29],
- c. disclaim liability for self and on part of the CA for any user who has not otherwise entered into an agreement with the user, and
- d. get the brand name (CN) of the CA to the end-user, so the end-user can start playing a part in the governance of the CA.

What should Auditors do?

Ask how to deliver value to end-users.

What should Standards Committees do?

The next great challenge for the security committees is to standardise the OODA loop.

Recommendations: others

- Vendors: direct audit to *their* end-users
- R/L/O: are vendor's liabilities clear?
- Auditors -- how to add value to Users not payers
- Standards: standardise the OODA loop

References

Thank You

[1] This presentation found its origins in an earlier briefing paper, "Audit Story," presented to the 2007 meeting of the (new) Board of CAcert.

[2] Microsoft, Inc., "Microsoft Root Certification Program," <http://technet.microsoft.com/en-gb/library/cc751157.aspx> Especially, note points 1, 3.

[3] Practically, the limitation to zero is achieved by the layering of many legal defences, not by some simple clause.

[4] A further result of this is that the CA can show an insurer that according to their legal defences, no claim will ever pay out, and thus they are a good risk for insurance.

[MRCP] Microsoft, op cit.

[5] a20070921.2. Ruling at <http://wiki.cacert.org/wiki/Arbitrations/a20070921.2>

[6] The CAcert story is somewhat unique, written by the members of the Association, and might not present a useful model to anyone else. Especially, "form an association" is not necessarily an absolute answer, as these carry a lot of costs in management, which might also cause collapse.

[7] My company, Systemics, implemented various Java and Perl implementations of PGP code and designs throughout the 1990s.

[8] CAB Forum, *EV Certificate Guidelines* Version 1.1

[9] See for example, "Improvements on Conventional PKI Wisdom," Carl Ellison 1st Annual PKI Research Workshop, which states "The third

element of conventional wisdom is that with a proper ID certificate, you can know the person with whom you are transacting. This idea traces back to the 1976 Diffie-Hellman paper [2], which make the assumption that the first important job was to learn the identity of the party on the other end of a communications channel."

[10] Reuters/Second Life, "[SL business sues for copyright infringement](#)" 3rd Jul, 2007 Also see the commentary in [Virtually Blind](#), Benjamin Duranske.

[11] CAcert, [Dispute Resolution Policy](#), cacert.org/policy/DisputeResolutionPolicy.php COD7.

[12] CAcert, <http://wiki.cacert.org/wiki/ArbitrationCases>.

[13] E.g., see The Federal Arbitration Act of the USA, as described briefly at wikipedia:
http://en.wikipedia.org/wiki/Federal_Arbitration_Act Also known as Section 9.

[14] CAcert Assurance Policy, <http://svn.cacert.org/CAcert/Policies/AssurancePolicy.html> DRAFT.

[NRP-DaL] CAcert, *Non-related person - Disclaimer and Licence*, www.cacert.org/policy/NRPDisclaimerAndLicence.php.

[DRP] CAcert, *Dispute Resolution Policy*, www.cacert.org/policy/DisputeResolutionPolicy.php.

[15] Minutes of the Advisory meeting 2007.08.18-19, svn.cacert.org/CAcert/Advisory/Minutes/AMinutes20070818.html point 1. Policy now at www.cacert.org/policy/OrganisationAssurancePolicy.php.

[16] This is somewhat restricted in the secure browsing and email worlds by audit and vendor criteria. Some criteria imply a verified Name, for some certificates. For example, [Mozilla CA Certificate Policy](#), pt 7-c.

[17] CAcert, [CPS 2.2. Publication of certification information](#). *work-in-progress* therefore not binding nor policy at time of writing.

[18] BBC, "The IRA campaigns in England," [website](#)

[19] Adi Shamir, Turing Award Lecture, 1st law of security, financialcryptography.com/mt/archives/000147.html

[20] Like all security *silver bullets*, the real price for these devices is not the sticker price -- around \$3000 to \$10,000 per -- but the project cost for the overall package. The total cost of ownership can be very high: the knowledge and work required to drive them, the knowledge required to certify them as secure (and certify any certifications they come with), the knowledge required to identify their weaknesses, and the work to address those in the wider security model, the hardware and systems that are required to deal with them, and the spares that need to be available quickly on failure.

[21] Stichting Oophaga, <http://oophaga.nl/> Stichting means (roughly) Foundation, Oophaga is a small frog. The 'ph' is pronounced hard.

[22] Board decision (m) 2006.11.23, svn.cacert.org/CAcert/CAcert_Inc/Board/board_review_actions_20040820_20070525.html

[23] For those interested, it is documented here:
http://svn.cacert.org/CAcert/CAcert_Inc/Funding/ Basic deal is 9000 euros over 4 tranches, each tranche splitting evenly between three budgets: retainer, expenses, and "work needed by CA" components. As of today, two of the tranches are paid by NLnet. For guarding

independence, there are safeguards built into the agreement.

[24] [Mozilla CA Certificate Policy](#), *ibid.*

[25] Attest Standard, AICPA

[26] This is not to say that other CAs were ignored. Instead, they were treated as a final sanity check; if their solutions looked like CAcert's solution, then the check passed. If they did not look like CAcert's solution then the difference had to be researched and understood before we moved on.

[27] *Audit directives*, <http://wiki.cacert.org/wiki/AuditDirectives> Note that this is a historical reconstruction.

[28] In directing the auditor, note that this does not mean either paying or entering into the contract. It instead means simply stating the requirements in criteria.

[29] In part they already do this, in their policies, it is simply that the label at the top of the page does not say "Criteria."