

Trusting Components

- People need to use components in development and need to access the services components offer. They need to be able to trust the behaviour of components and have ample reasons not to do so. There are no simple solutions to the dilemmas posed.
- Whether you are a service user, a service developer, an integrator, a supplier of security or trust related products, you need to understand the patterns of trust and be able to detect and respond to breaches of trust. To believe otherwise is to invite trouble.
- Our basic premise is that certain patterns of relationships allow trust to be developed in a virtuous circle and certain patterns look more like sitting targets waiting for trouble. To allow you to develop your own insights into how this affects your business we supply a set of services:
- As with any complex problem, the first step is to explore the terrain. We suggest you contact us for an initial exploration of how we might move your business forwards.

- Assessment / Audit
- Design / Design Review
 - Component / Web Service
 - Whole System / Architecture
- Market Survey
- Process Improvement / Review
- Training / Coaching / Mentoring

Veryard Projects is a leading player in the architecture of component systems and their emergent properties.

Antelope Projects is one of very few consultancies supporting clients in the management of business risk in their networks of stakeholders.

We have teamed up to support clients in developing effective approaches to doing business based on component networks. We deal with software component networks or all scales and we deal with business component networks.

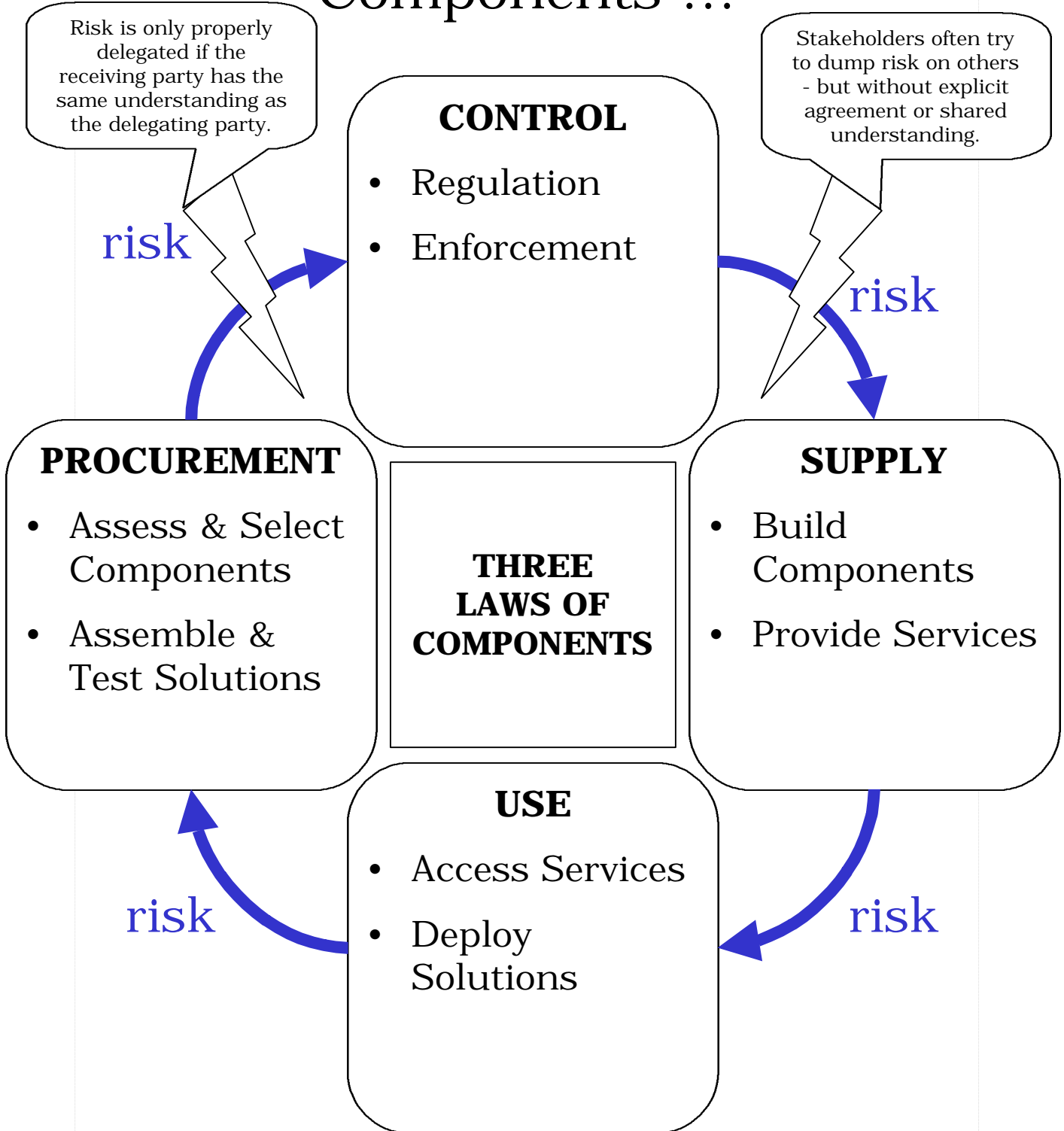
Three Laws of Robotics (Isaac Asimov)

1. A robot may not injure a human being, or, through inaction, allow a human to come to harm.
2. A robot must obey orders given to him by human beings except where such orders would conflict with the First Law.
3. A robot must protect its own existence as long as such protection does not conflict with the First or Second Law.

Three Laws of Components (CBDi Forum)

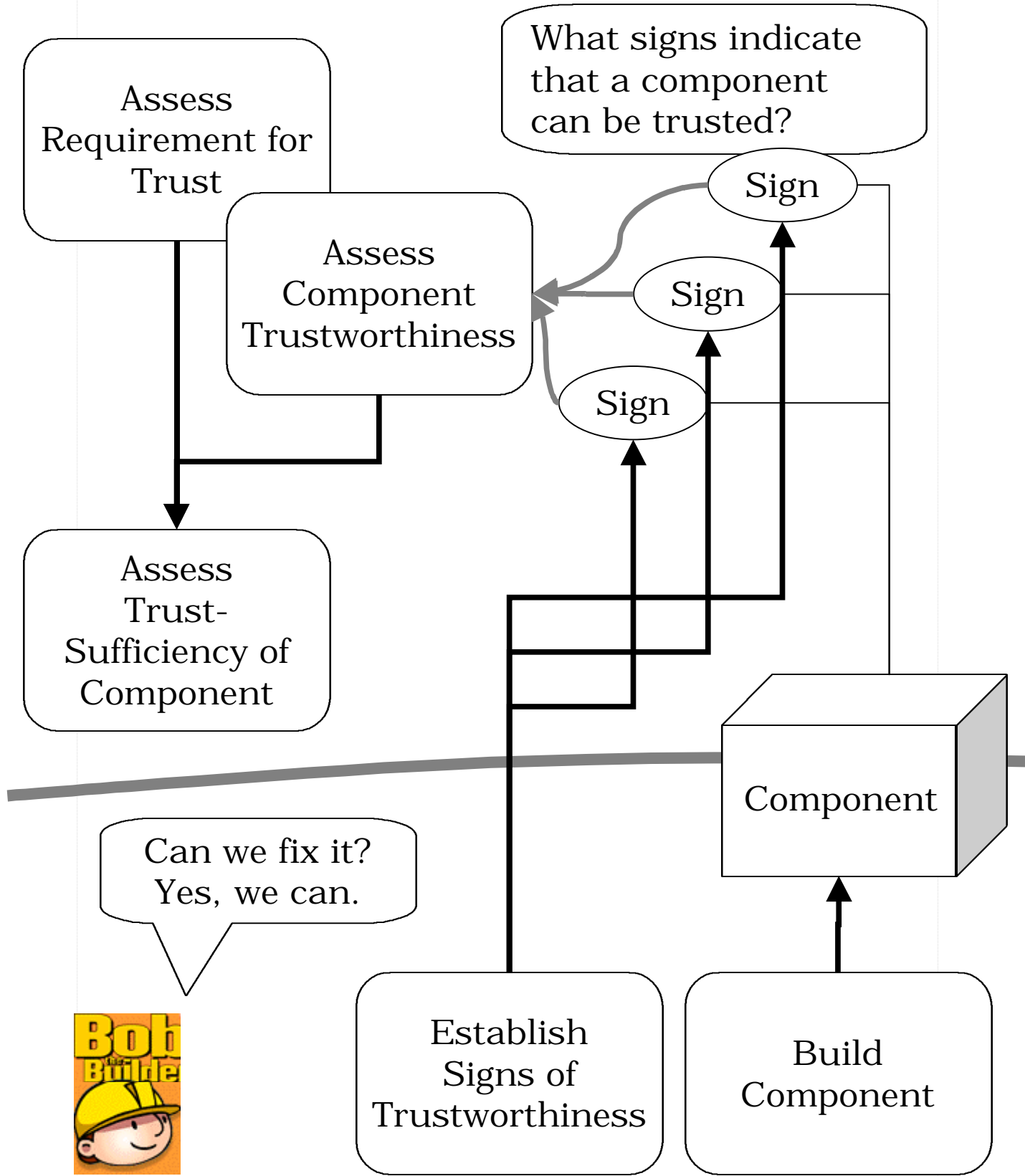
1. A component must be safe to use - it doesn't introduce any new or incremental risks for the user - whether privacy or otherwise.
 - No side effects, no leaks, no feature interaction.
2. A component must deliver the services demanded of it.
 - This is availability in the broadest sense, although this overlaps with 1.
3. A component must look after itself. It must be viable, it must protect itself from harm.
 - Viability includes such notions as robustness, economics of scale (aka reuse) and biodiversity. Note that an attack on a component matters primarily because it undermines 1 and 2.

Processes for Trusting Components ...

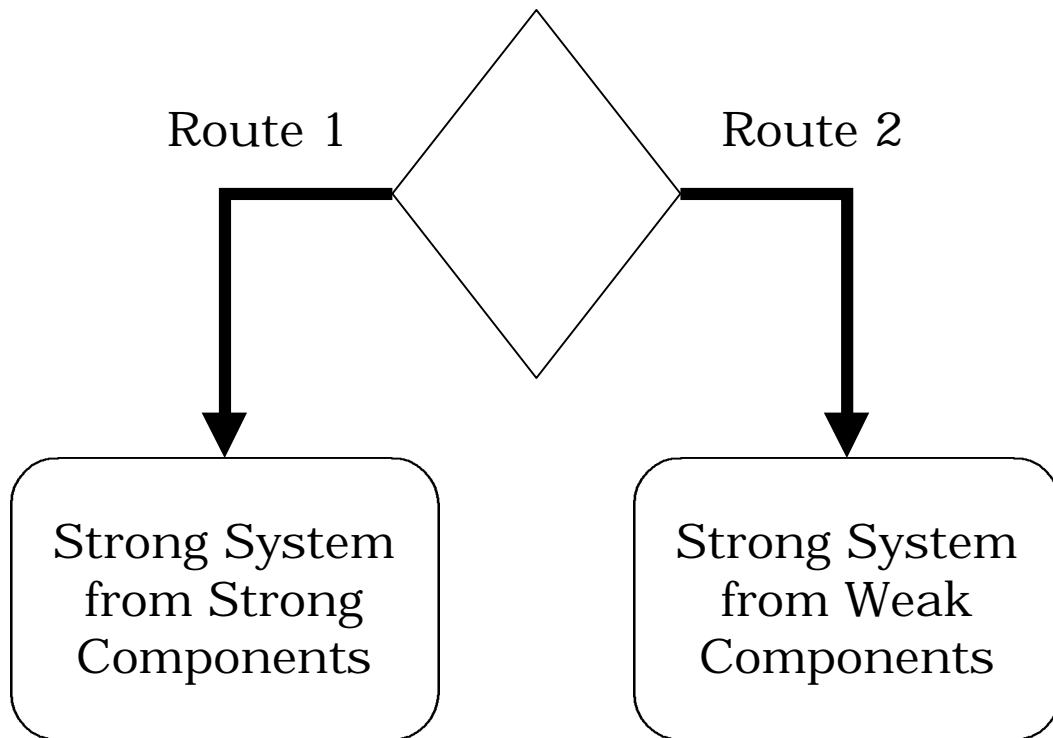


...and for Delegating Risk

Context: Component Selection



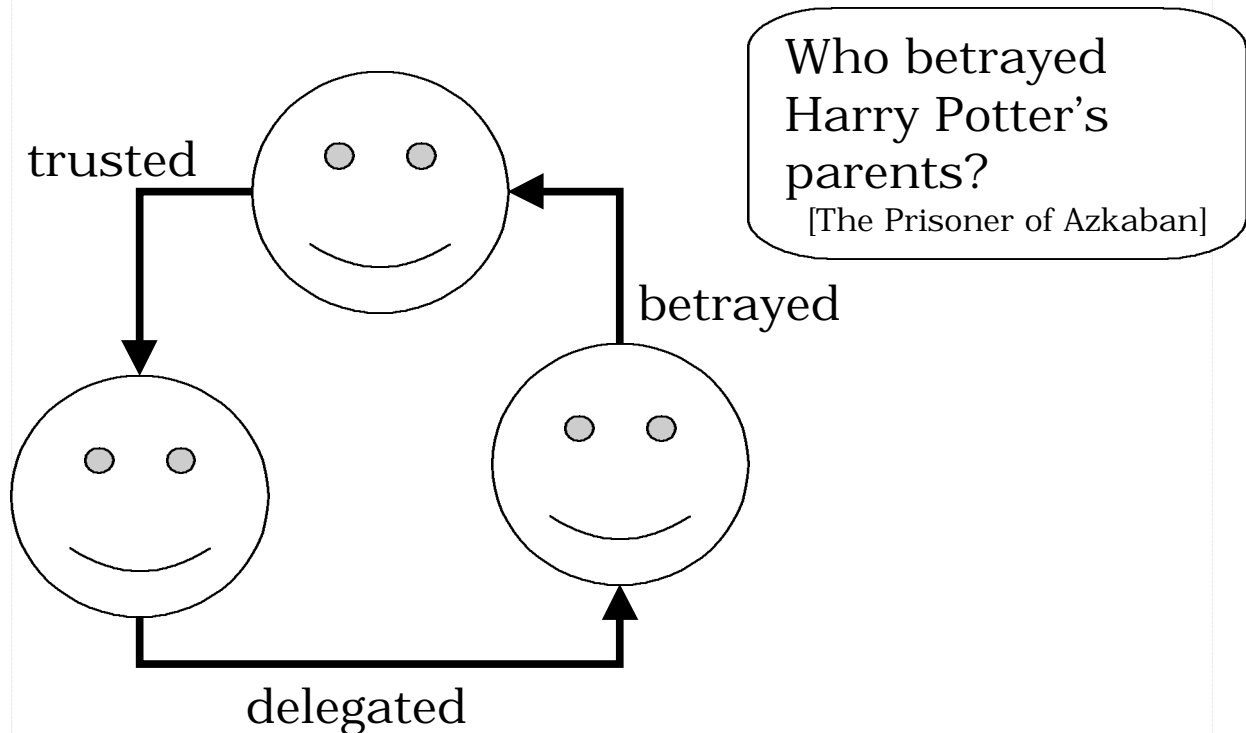
Context: System Design Strategy



- Hero components vulnerable to attack - single point of failure - high gain from attack
- Brittle - major consequences from system failure.

- System strength is emergent property - not dependent on any single component.
- Robust, evolutionary, self-healing network.
- Doesn't happen by chance - needs appropriate patterns.

Networks of Trust



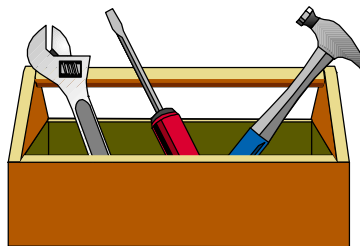
Paradoxes of Trust

- The strongest component is the most obvious place to attack.
- Strength that depends on secrecy is always vulnerable to leakage.
- Fixed source of strength is always vulnerable to erosion.
- A chain of trust is as strong as its weakest link.
- ...

Further reading & other sources



- Bruce Schneier, ***Secrets and Lies*** (John Wiley, 2000)
- Richard Veryard, ***Component-Based Business: Plug and Play*** (Springer Verlag, London, 2001)



- CBDi Forum
 - <http://www.cbdiforum.com>

Available Services

- Assessment / Audit
- Design / Design Review
 - Component / Web Service
 - Whole System / Architecture
- Market Survey
- Process Improvement / Review
- Training / Coaching / Mentoring

Contact Details

Antelope Projects

Aidan Ward

- Phone: 01474 702259
- aidan_ward@antelopes.com

Veryard Projects

Richard Veryard

- Phone: 020 8810 4431
- richard@veryard.com