Trusted Computing and Provenance:

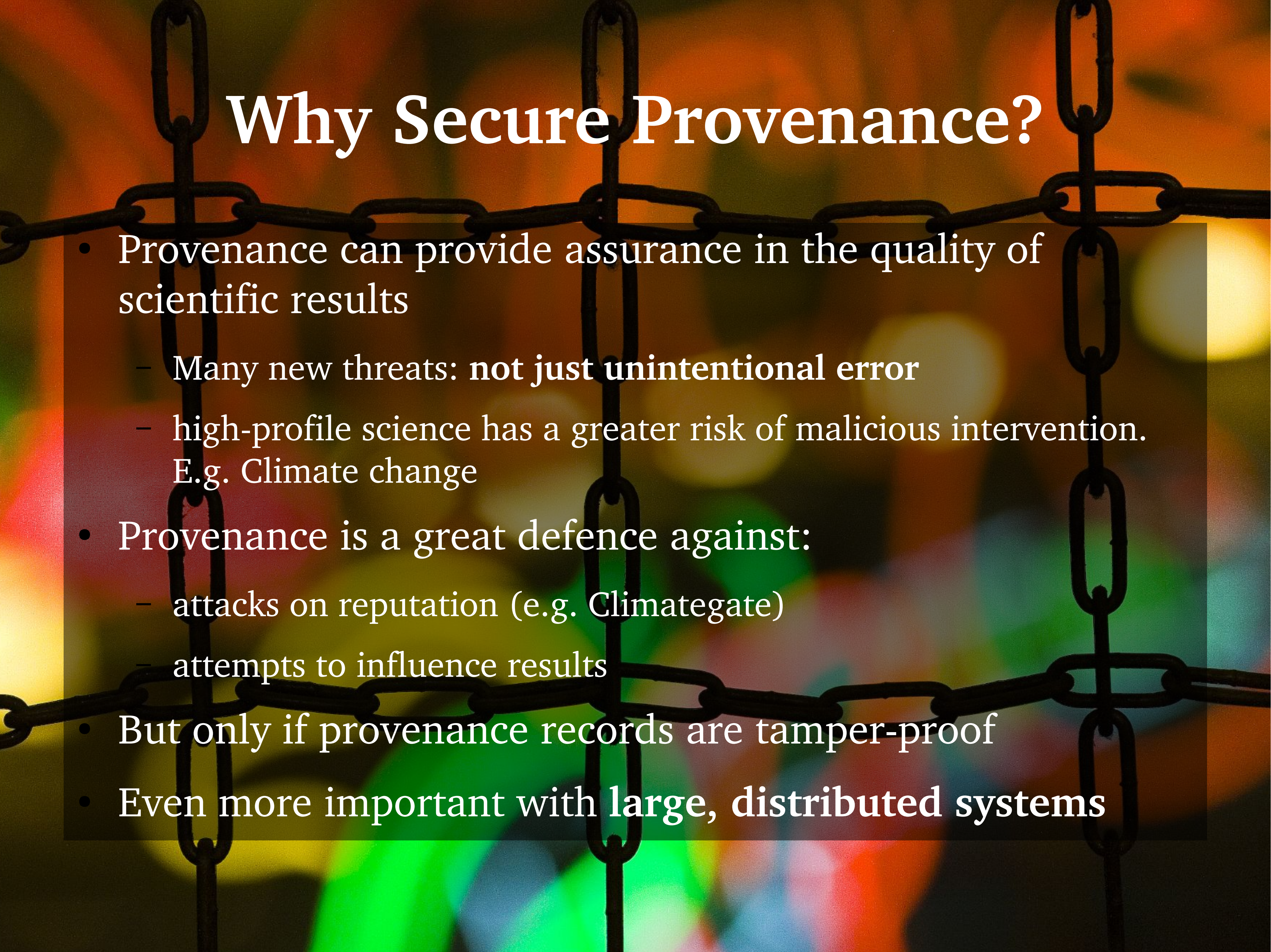
Better Together

John Lyle and Andrew Martin
Oxford University Computing Laboratory

Some Background

- My research is in security, not provenance (sorry!)
- We're interested in **assurance** of platform **behaviour** through reporting **system state**
- Part of provenance is knowing **system state** to support and guarantee consistent **behaviour**
- Growing interest in **secure** provenance

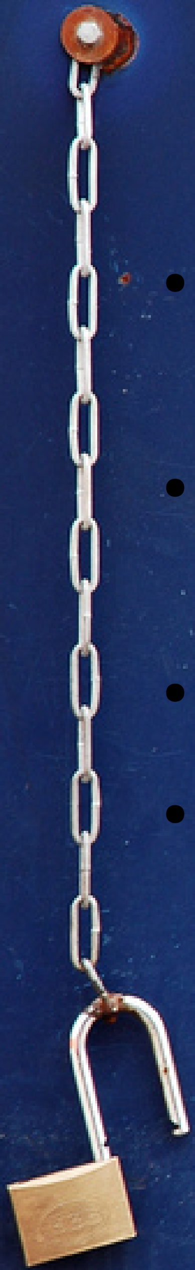
- Surely there's some overlap...



Why Secure Provenance?

- Provenance can provide assurance in the quality of scientific results
  - Many new threats: **not just unintentional error**
  - high-profile science has a greater risk of malicious intervention. E.g. Climate change
- Provenance is a great defence against:
  - attacks on reputation (e.g. Climategate)
  - attempts to influence results
- But only if provenance records are tamper-proof
- Even more important with **large, distributed systems**

Trusted Computing

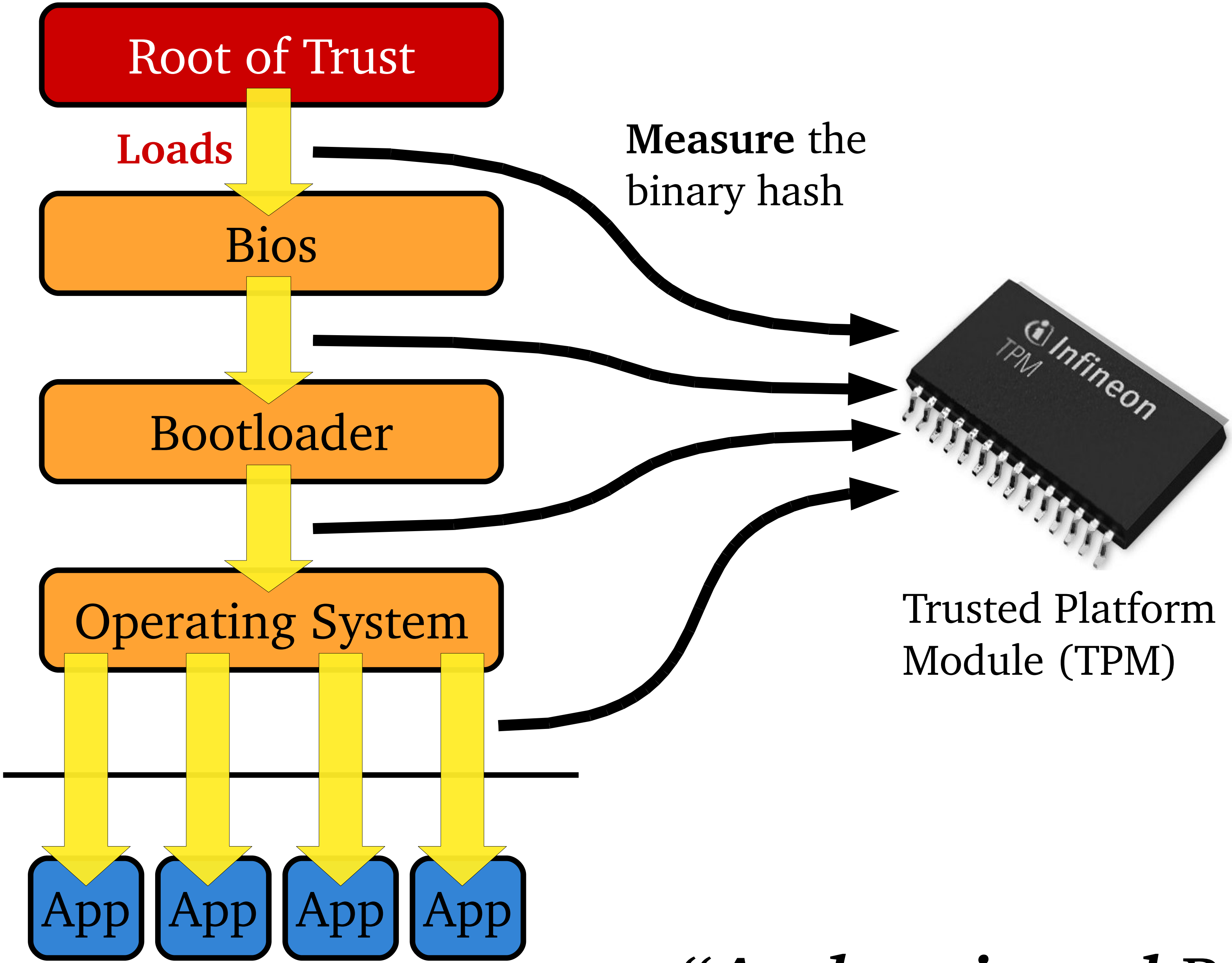


- Trusted Computing can provide tamper-proof **guarantees** of program execution.
- It can provide information about **hardware** and **software**
- To explain how, need to go into some technical details
- Stop and ask me questions!
  - Time constraints mean I'm glossing over lots of details.

Integrity Reporting

Assessing trustworthiness by asking:

“What programs are you running?”



Root of Trust

Loads

Bios

Bootloader

Operating System

App

App

App

App

Measure the binary hash

Trusted Platform Module (TPM)

“***Authenticated*** Boot”

```
PCR   Hash value                                Executable

10    61a3393ccbabc6e6fd16809b105eba9737779c70   boot_aggregate
10    42d55319f874a2c6c39c2afc04ce38f177000a60   fan        [kernel module]
10    d9f54f7f0a296ae15a542e0a4110f1b8cbed9c9c   processor  [kernel module]
10    9d4f0f315936756c83cf497b4c41e3cf26df408a   thermal    [kernel module]
...
10    df20bd67ea041bcb2f535b823a876e6540efaf12   /bin/sh
10    68212426a0ede03a51db098cb251dc9b5d1c2bc9   /bin/mount
10    007857e17791383d2d6c6945b325cb05c0b152df   /bin/bash
10    932cbb260bca27d29a6cb0cf7e422cefe58f6f93   /bin/mknod
10    5851490d5ab05c3457cebab87d1f59aa8a76fc66   /bin/ln
10    88d06c92e771012e449d1c72f30ba4c4950b286d   /bin/mkdir
10    1f5b13cdc44667b934e77026cf71233bc7ab4893   /bin/grep
10    d16a079245e5d37539daed12734af5c209fc5290   /bin/cp
10    53b00417eccbdd21d382c998b49b91461228e2c8   /usr/bin/find
10    dbb2d4f21f83ec9cbe6b52cd912e0bf8eae94e60   /lib/libm-2.8.so
10    c6a2fb35500e3fac614abf4bcd6bfcad660619f4   /bin/chmod
10    7e2876fc66bb168fc166d6f9c0b9a7f956090366   /bin/hostname
10    5f463c6051608d346f881fbb317e32bb86d99bec   /bin/login
10    cd4d0efb740193fabfa496a2d2368a4d1724c3de   /lib/libshadow.so.0.0.0
10    b24c85124cd83edf8ec8505d1a5f0f8a0d9cd2b9   /lib/libpam_misc.so.0.81.3
10    a458983560e7487fcfb175140f1232f68a1d257d   /lib/security/pam_securetty.so
...
```

$$\mathbf{PCR10 = SHA1(\ A_n,\ SHA1(\ ...\ SHA1(\ A_1\ ,\ SHA1(A_0,\ 0x00\ ))))}$$

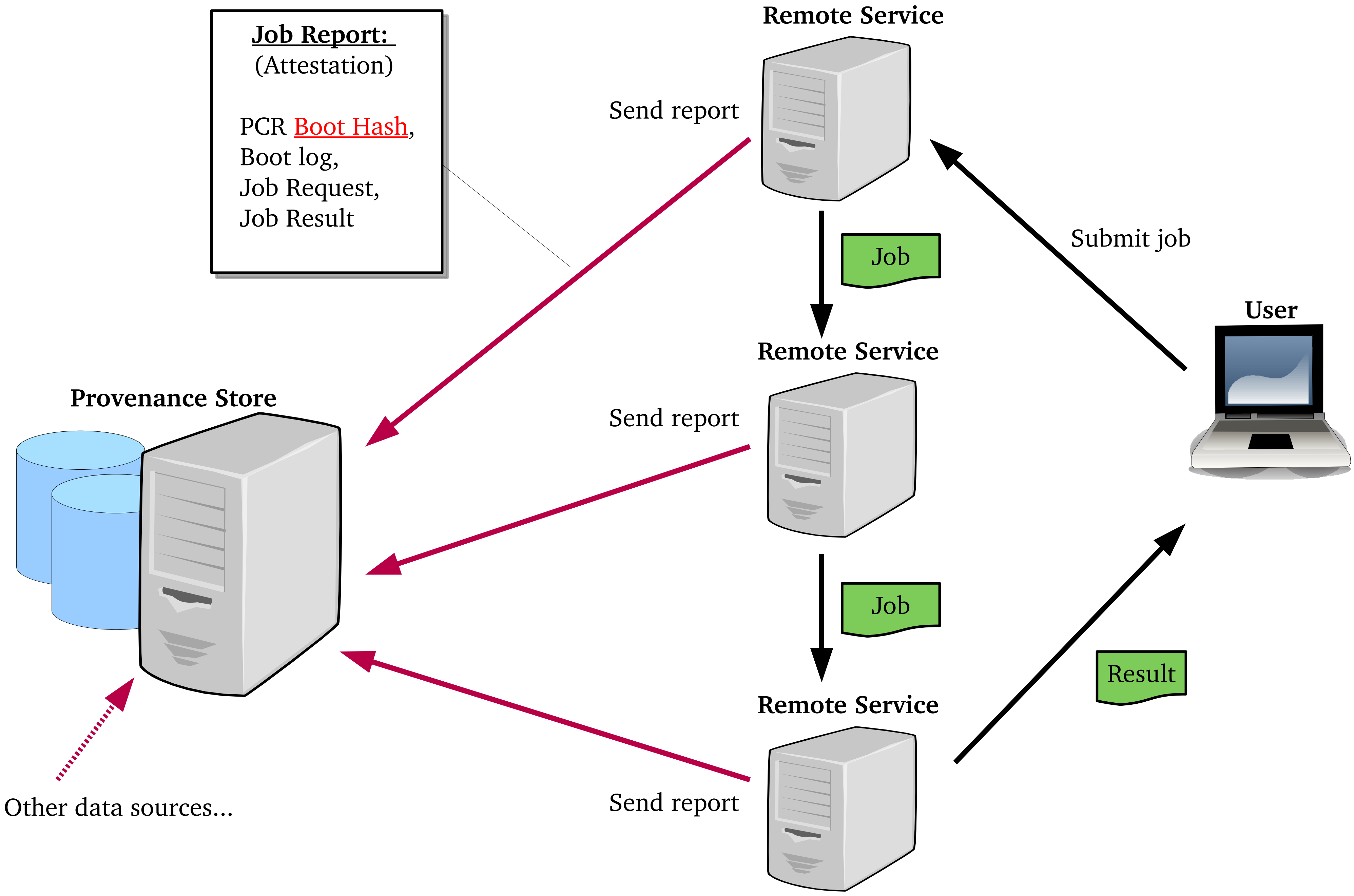
Remote Attestation

Sign *a* copy of your boot measurements

What about Provenance?

- Security and provenance rely on establishing a complete picture of the factors influencing a remote computer's behaviour
- Trusted Computing can do it in a **tamper-resistant** manner
- Attestations can be considered *trustworthy actor-state p-assertions*
- This is immediately applicable to large-scale grid computing.
- We have the technology already!

Attestation-based Provenance



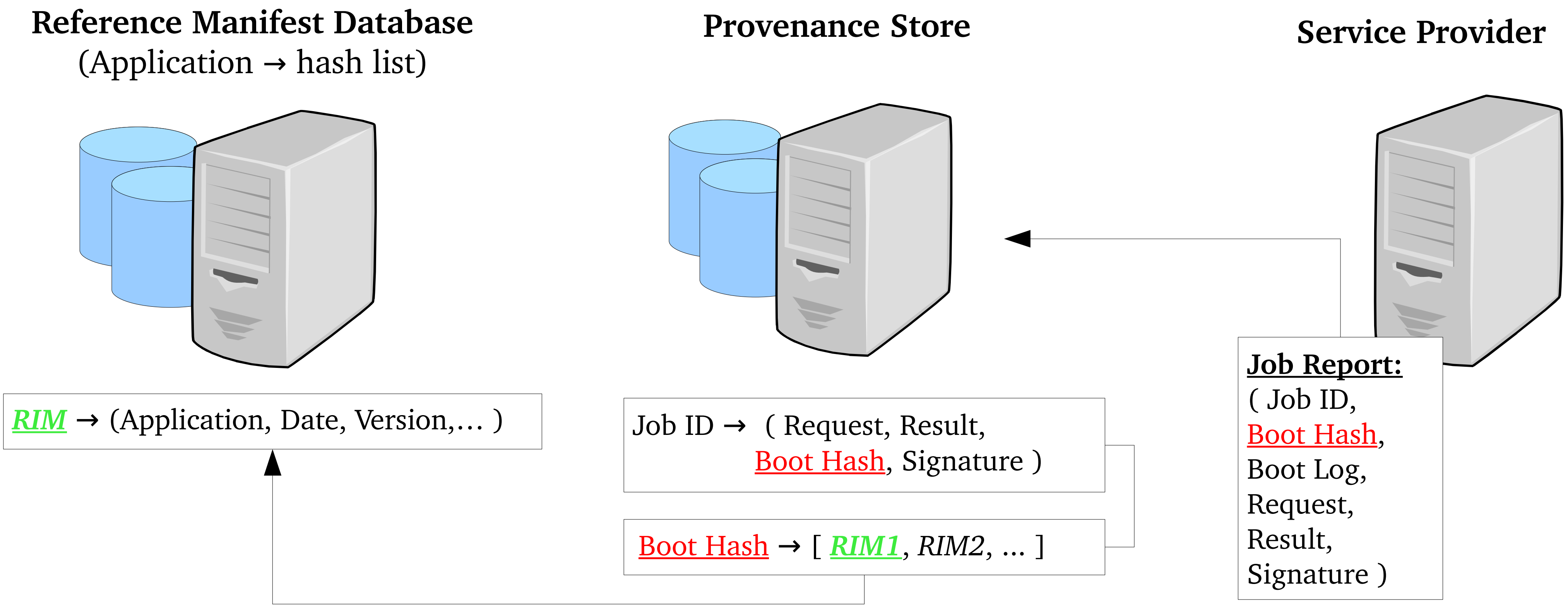
Job Report:
(Attestation)

PCR Boot Hash,
Boot log,
Job Request,
Job Result

Information Collected

- Platform unique identity (AIK)
- All software identities
  - Firmware, drivers, operating systems, applications
- Hardware identities*
- Timestamps
- Job information**
  - A hash of the job / request message
  - A hash of the calculated result

Optimising Storage



Reference Manifest Database
(Application → hash list)

Provenance Store

Service Provider

RIM → (Application, Date, Version,…)

Job ID → (Request, Result,
Boot Hash, Signature)

Boot Hash → [*RIM1*, *RIM2*, …]

Job Report:
(Job ID,
Boot Hash,
Boot Log,
Request,
Result,
Signature)

We Have The Technology...

- Software for Java, C++, .net
- TPMs are cheap and available
- Linux has native support for Authenticated Boot and TPMs. Windows too*
- Just needs to be integrated into middleware
- Virtualisation makes much of this easier
    - Report on a virtual machine image

What we can't do (yet)

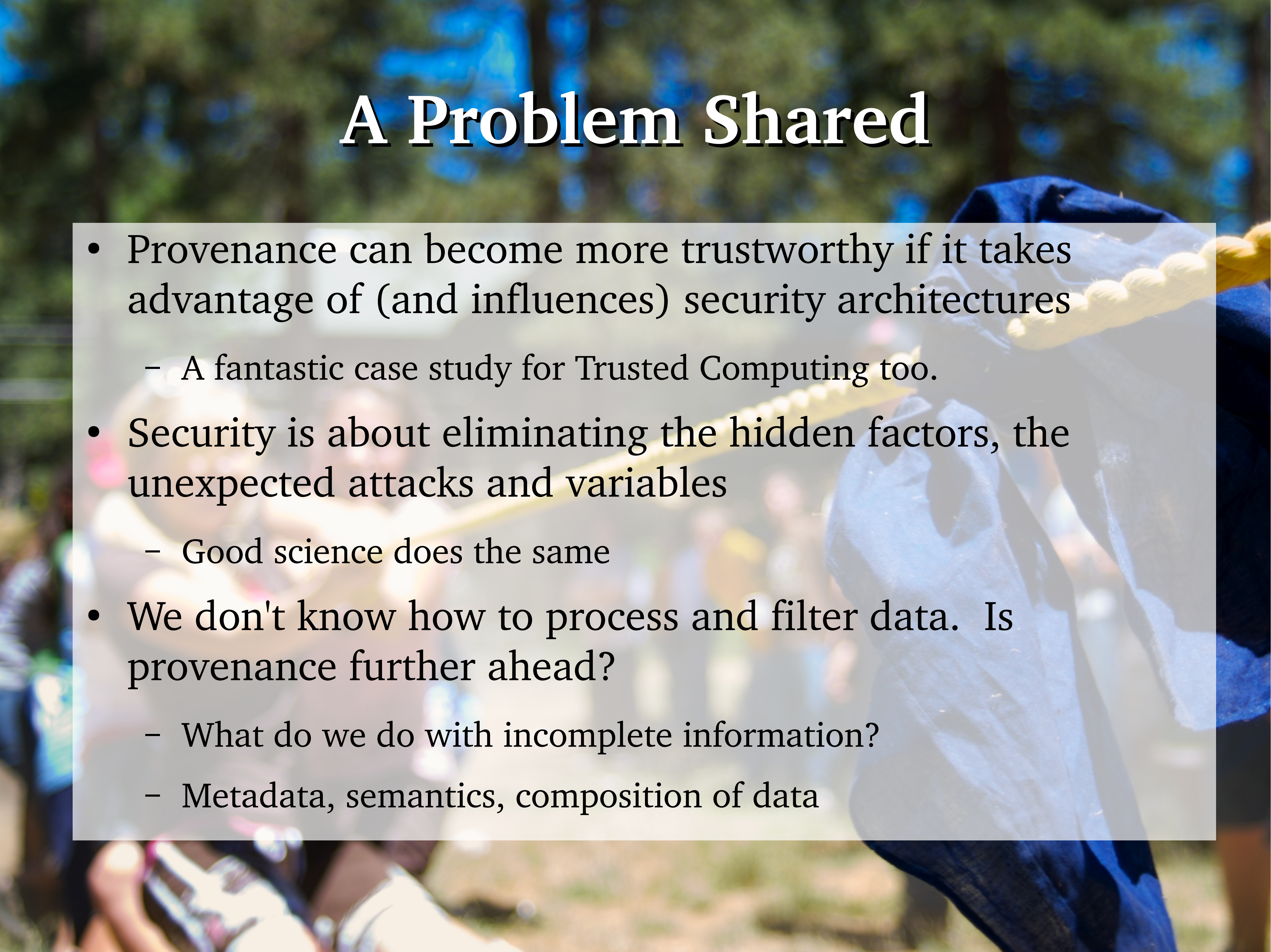
- Runtime information and configuration details
  - Can be added, but needs some work
  - This is some of my future work
- Needs **integrating** with other provenance information
  - Purpose of experiment, sources of data, etc...
- **Recreating** results is not an **automatic** process
  - Virtual machines may also help here
- Need to have a frequently-updated **software database** (RMDB)

Part 2:

Better **together?**

Research in Common

- Trusted Computing and provenance have a lot of common research.
- Secure, transparent logging
- Usage control / monitoring
- Compilation histories
- Secure storage
- Even using the same examples
  - Grid, SOA, cloud
- Desire to automate and scale
- **Integrity!**



A Problem Shared

- Provenance can become more trustworthy if it takes advantage of (and influences) security architectures
  - A fantastic case study for Trusted Computing too.
- Security is about eliminating the hidden factors, the unexpected attacks and variables
  - Good science does the same
- We don't know how to process and filter data. Is provenance further ahead?
  - What do we do with incomplete information?
  - Metadata, semantics, composition of data

… is a problem doubled?

- Different **research directions**.
  - Cryptographic strength vs data consistency and accuracy
- Lots of new and interesting security challenges, maybe Trusted Computing wont help with the big ones?
- How do we **develop secure software**?
  - If grid middleware is vulnerable to runtime attack, have we gained anything?
- Other issues: **PKI**, **performance**, **usability**, **privacy** …
- My literature review just became twice as long!

Conclusion

- Two fields that are solving similar problems
  - We both want tamper-proof identification of systems
- There is a lot of **immediately applicable** software and hardware
  - **exciting opportunity** for researchers and developers of provenance tools.
- If we work together, **trusted provenance** shouldn't be that far away.