The imagination driving Australia's ICT future.

NATIONAL ICT AUSTRALIA LIMITED

Trustworthy Embedded Systems
How Australian Research Can Have an Impact

Gernot Heiser

Program Leader — Embedded, Real-Time and Operating Systems — NICTA
Professor of Operating Systems — UNSW
Founder and CTO — Open Kernel Labs, Inc

gernot@ok-labs.com

Australian Government
Department of Communications, Information Technology and the Arts
Australian Research Council

NICTA Members

THE AUSTRALIAN NATIONAL UNIVERSITY

UNSW

First for Business
Department of State and Regional Development

BusinessACT
ACT GOVERNMENT

Victoria
The Place To Be

THE UNIVERSITY OF MELBOURNE

The University of Sydney

Queensland Government

Griffith UNIVERSITY

QUT
Queensland University of Technology

THE UNIVERSITY OF QUEENSLAND
AUSTRALIA

NICTA Partners

What Are Embedded Systems?

Embedded Operating Systems

- The operating system is the lowest layer of software, interacting directly with the hardware

Embedded Operating Systems

- The operating system is the lowest layer of software, interacting directly with the hardware
- There are at least 50 embedded operating systems

Embedded Operating Systems

- The operating system is the lowest layer of software, interacting directly with the hardware
- There are at least 50 embedded operating systems
- The world needs another one like a fish needs a bicycle

Embedded Operating Systems

- The operating system is the lowest layer of software, interacting directly with the hardware
- There are at least 50 embedded operating systems
- The world needs another one like a fish needs a bicycle

WRONG!

Modern Embedded Software is Big and Complex

Examples:

- Mobile phone handsets run software that consists of *millions* of lines of code
- A top-of-the-line car has in excess of a Gigabyte of software

Modern Embedded Software is Big and Complex

Examples:

- Mobile phone handsets run software that consists of *millions* of lines of code
- A top-of-the-line car has in excess of a Gigabyte of software

The consequence?

Modern Embedded Software is Big and Complex

Examples:

- Mobile phone handsets run software that consists of *millions* of lines of code
- A top-of-the-line car has in excess of a Gigabyte of software

The consequence?

Modern Embedded Software is Big and Complex

Examples:

- Mobile phone handsets run software that consists of *millions* of lines of code
- A top-of-the-line car has in excess of a Gigabyte of software

The consequence?

Modern Embedded Software is Big and Complex

Examples:

- Mobile phone handsets run software that consists of *millions* of lines of code
- A top-of-the-line car has in excess of a Gigabyte of software

The consequence?

Modern Embedded Software is Big and Complex

Examples:

- Mobile phone handsets run software that consists of *millions* of lines of code
- A top-of-the-line car has in excess of a Gigabyte of software

The consequence?

Other Scary Examples

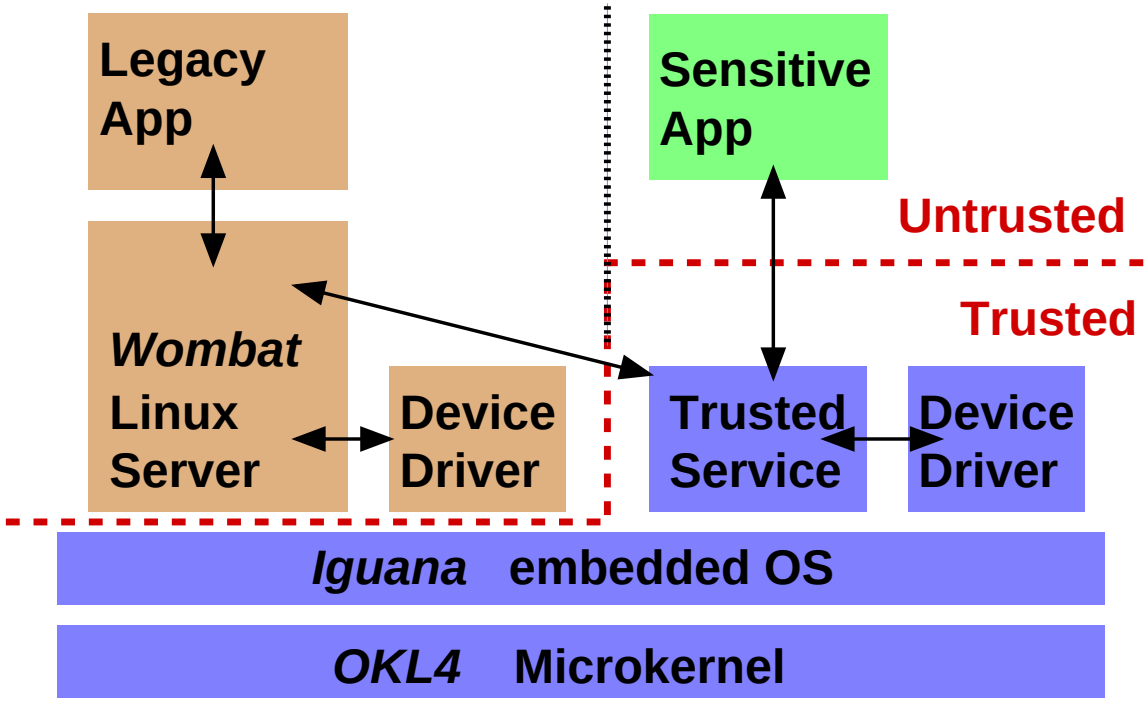
- Internet banking
  - ➜ how safe are your access keys?
  - ➜ how safe is your money?
- Health cards
  - ➜ how private is your health data?
  - ➜ is someone changing your medication?
- Property protection
  - ➜ is someone disabling your house/car alarm?
- Personal safety
  - ➜ is some hoon hacking their steering/breaks?
  - ➜ is someone hacking *your* steering/breaks?
- Mobile communication infrastructure
  - ➜ will your phone become a jammer?
  - ➜ will 100,000 phones become jammers?

Operating Systems Can Help!

OKL4/Iguana: An Operating System for Trustworthy Embedded Systems

OKL4/Iguana:

- supports structuring embedded software into isolated components
- allows running an other operating system on top (eg Linux, Windows)

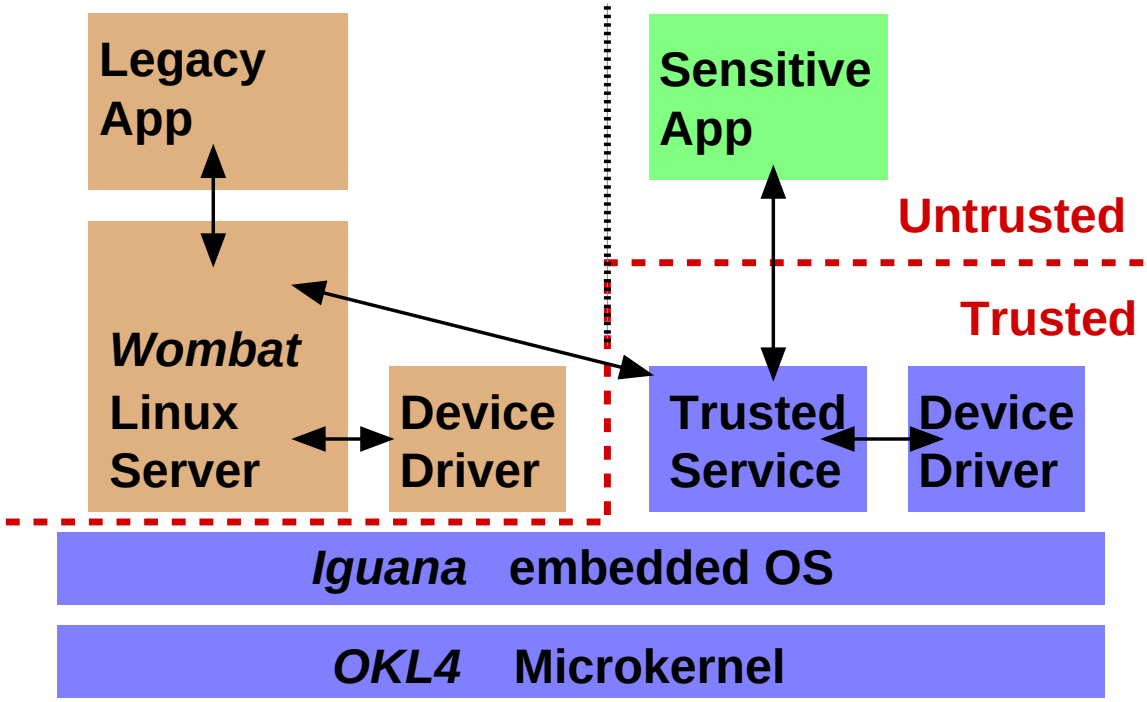


Operating Systems Can Help!

OKL4/Iguana: An Operating System for Trustworthy Embedded Systems

OKL4/Iguana:

- ➜ supports structuring embedded software into isolated components
- ➜ allows running an other operating system on top (eg Linux, Windows)
- ➜ minimises the amount of code that needs to be trusted (as little as 20,000 lines of code)

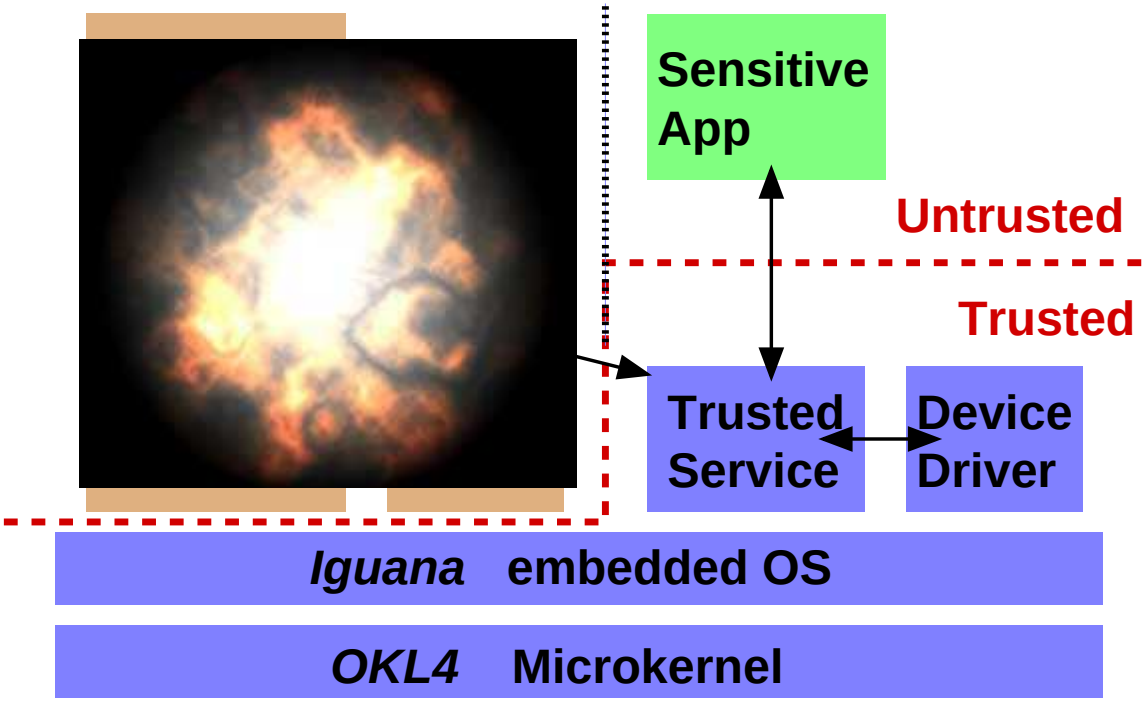


Operating Systems Can Help!

OKL4/Iguana: An Operating System for Trustworthy Embedded Systems

OKL4/Iguana:

- ➜ supports structuring embedded software into isolated components
- ➜ allows running an other operating system on top (eg Linux, Windows)
- ➜ minimises the amount of code that needs to be trusted (as little as 20,000 lines of code)
- ➜ protects critical components from others

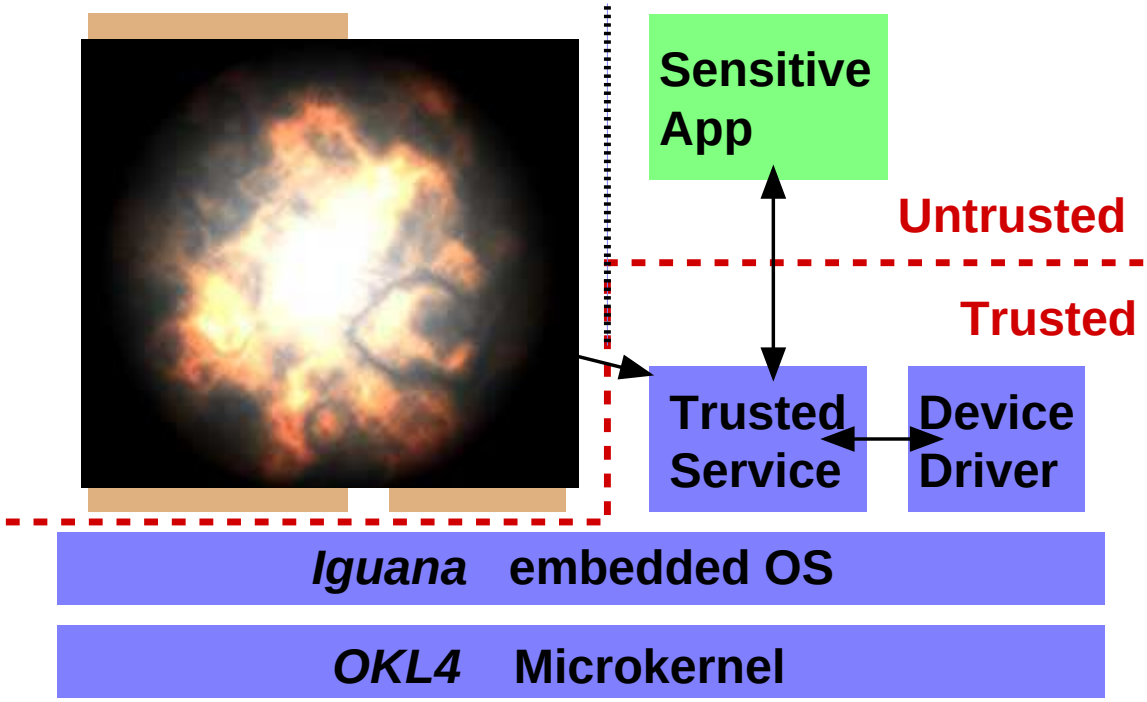


Operating Systems Can Help!

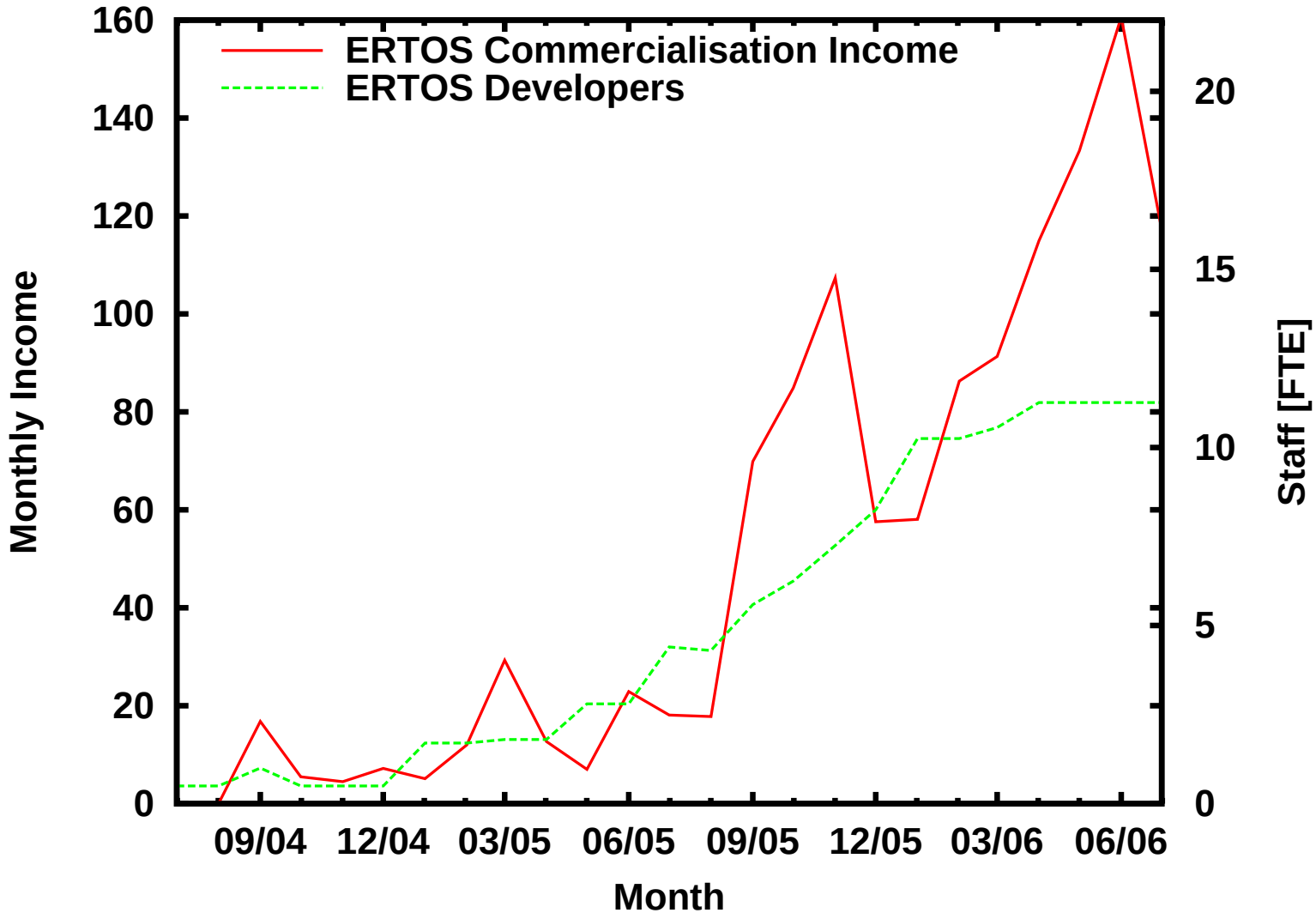
OKL4/Iguana: An Operating System for Trustworthy Embedded Systems

OKL4/Iguana:

- ➜ supports structuring embedded software into isolated components
- ➜ allows running an other operating system on top (eg Linux, Windows)
- ➜ minimises the amount of code that needs to be trusted (as little as 20,000 lines of code)
- ➜ protects critical components from others
- ➜ *all open source!*



COMMERCIALISING OPERATING SYSTEMS



NATIONAL ICT AUSTRALIA LIMITED

Customers and Partners

February 2004	ST Microelectronics
August 2004	Qualcomm
September 2005	Multinational 3
July 2006	Multinational 4
August 2006	Ericsson
October 2006	Multinational 5
Q1 2007	first mobile phone in stores
Q3 2007	first non-phone consumer item

Vision: Phase I

Maximise market penetration

- Based on our present technology (and incremental improvements)
  - ➜ service-based business
  - ➜ proceeds fund on-going development
- Become industry standard in a number of verticals
  - ➜ enabled by openness, superior technology and lower cost
  - ➜ on track for mobile phone handsets
  - ➜ entries made into other sectors
- Ecosystem of SME partnerships
  - ➜ local companies with complementary products

Vision: Phase I

Maximise market penetration

- Based on our present technology (and incremental improvements)
  - ➜ service-based business
  - ➜ proceeds fund on-going development
- Become industry standard in a number of verticals
  - ➜ enabled by openness, superior technology and lower cost
  - ➜ on track for mobile phone handsets
  - ➜ entries made into other sectors
- Ecosystem of SME partnerships
  - ➜ local companies with complementary products

Continue research in NICTA

- Leverage small size of system to achieve unprecedented reliability
  - ➜ complete and guaranteed timing model of kernel
  - ➜ formally-verified implementation — proven bug-free

Vision: Phase II

Disruptive technology changes the game

- Formal verification opens new markets
  - ➜ reduce certification cost (mil/aero/auto)
  - ➜ certification bodies likely to increase requirements (including for CE)
- Massive technology advantage strengthens competitiveness
  - ➜ achieve robustness beyond competitor's reach
- Transform into product business
  - ➜ licensing, subscriptions

What is Behind?

1. Superior Technology:

- 10+ years of research (originally at UNSW)
  - ➜ resulted in fastest and smallest system of its kind
- Unrivalled research agenda
  - ➜ 50 people in NICTA across 4 programs in 3 labs
  - ➜ strong collaboration, unique collection of skills
  - ➜ ability to out-innovate any competitors
- Goal: Unprecedented level of trustworthiness
  - ➜ mathematical proof of system correctness
  - ➜ migration path from present system

What is Behind?

2. Vision and Stubbornness:

- We want to change the world!
  - ➜ long-term goal
  - ➜ drives research agenda
  - ➜ motivates team

What is Behind?

2. Vision and Stubbornness:

- We want to change the world!
  - ➜ long-term goal
  - ➜ drives research agenda
  - ➜ motivates team
- Challenge: How to escape the bean counters?
  - ➜ successes are rare for a long time
  - ➜ hard to survive in publish-or-perish system
  - ➜ has all but killed off university research in operating systems

What is Behind?

3. The Best Team:

- 90% of spinout team are former UNSW students
  - ➜ universities are full of very bright kids
  - ➜ believing in their abilities is rewarding
  - ➜ challenge is to get their attention
  - ➜ challenging teaching is the key

What is Behind?

3. The Best Team:

- 90% of spinout team are former UNSW students
  - ➜ universities are full of very bright kids
  - ➜ believing in their abilities is rewarding
  - ➜ challenge is to get their attention
  - ➜ challenging teaching is the key
- Researchers come from all over the world
  - ➜ attracted by NICTA and what it offers

What is Behind?

4. The NICTA Model:

- Critical mass
  - ➜ ability to tackle projects requiring significant resources

What is Behind?

4. The NICTA Model:

- Critical mass
  - ➜ ability to tackle projects requiring significant resources
- Use-inspired basic research
  - ➜ encourages a vision of long-term impact
  - ➜ ... while ensuring that it is *really* research
- Strong encouragement and support for cross-area collaboration
  - ➜ develop strong synergies

What is Behind?

4. The NICTA Model:

- Critical mass
  - ➜ ability to tackle projects requiring significant resources
- Use-inspired basic research
  - ➜ encourages a vision of long-term impact
  - ➜ ... while ensuring that it is *really* research
- Strong encouragement and support for cross-area collaboration
  - ➜ develop strong synergies
- Encouragement and infrastructure for commercialisation
  - ➜ provides credible path to real-world impact
  - ➜ entrepreneur-in-residence program rocks!

What is Behind?

5. Open Source:

- It's a *disruptive mechanism* for spreading *disruptive technology*
  - ➜ destroys competitors' revenue models
  - ➜ eases and speed market penetration
  - ➜ replaces expensive sales and marketing teams
  - ➜ removes incentive for launching new competitors

What is Behind?

5. Open Source:

- It's a *disruptive mechanism* for spreading *disruptive technology*
  - ➜ destroys competitors' revenue models
  - ➜ eases and speed market penetration
  - ➜ replaces expensive sales and marketing teams
  - ➜ removes incentive for launching new competitors
- It's the *way of the future* for infrastructure technologies
  - ➜ openness is *essential* for trustworthiness
  - ➜ sharing cost of infrastructure makes economic sense

What is Behind?

1. Superior technology through long-term research
2. Vision and stubbornness
3. The best team, fed by UNSW student pipeline
4. The NICTA model of research and commercialisation
5. Open Source, the big disrupter