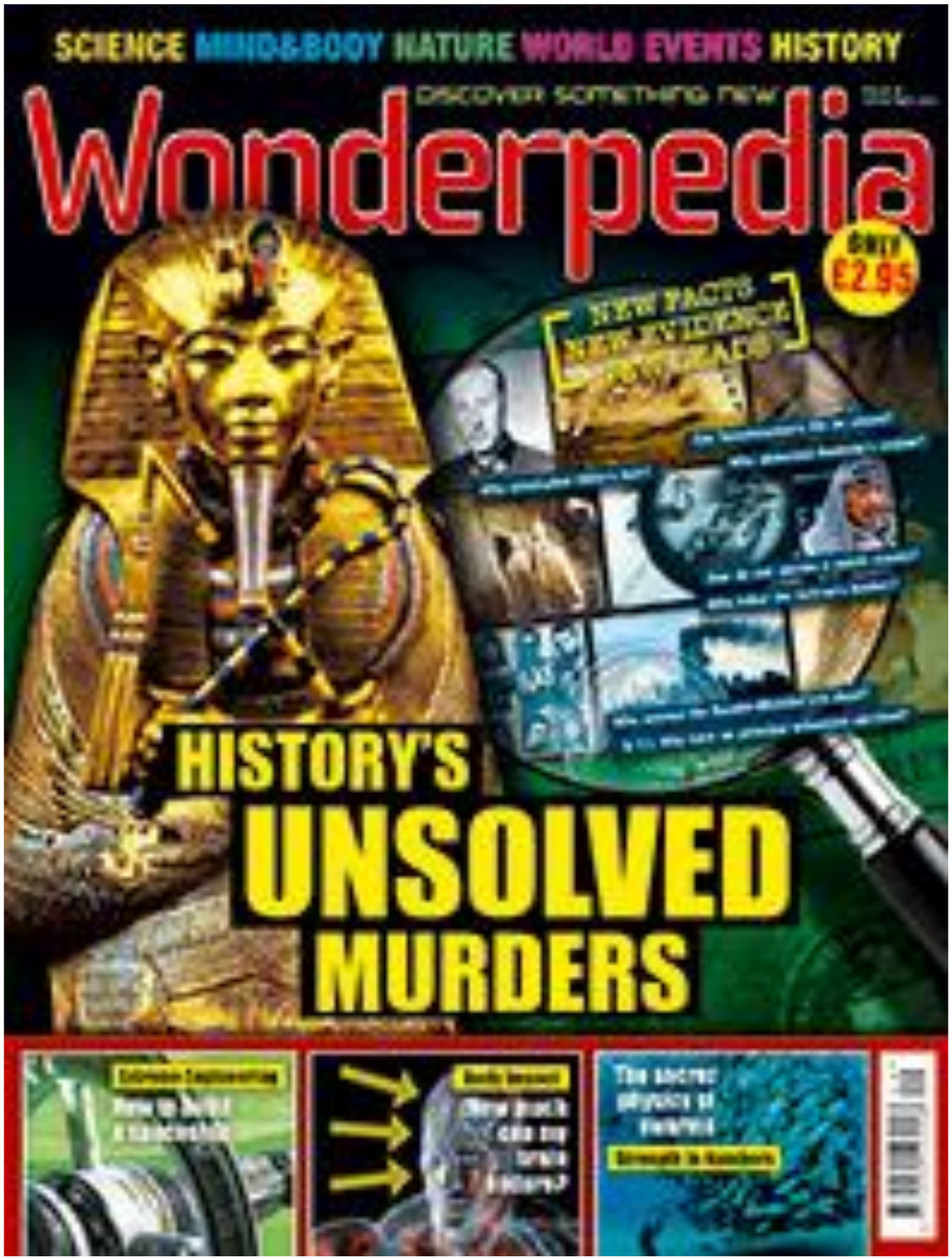


The Open University

Dirty business

Do germs have germs?

The Open University

Part I

RUNTIME TRACEABILITY FOR AN ADAPTIVE TIME MACHINE

Yijun Yu

E-type programs

- S-programs are written according to an exact specification of what the program can do

- *P*-programs are written to implement certain *procedures* that completely determine what the program can do (e.g., a chess game)

- ***E*-programs** are written to perform some real-world activity; how they should behave is strongly linked to the *environment* in which they run, and these programs need to adapt to varying requirements and circumstances in that environment

Lehman, Meir M. (1980). "**Programs, Life Cycles, and Laws of Software Evolution**". *Proc. IEEE* 68 (9): 1060–1076.

Lehman's law on complexity

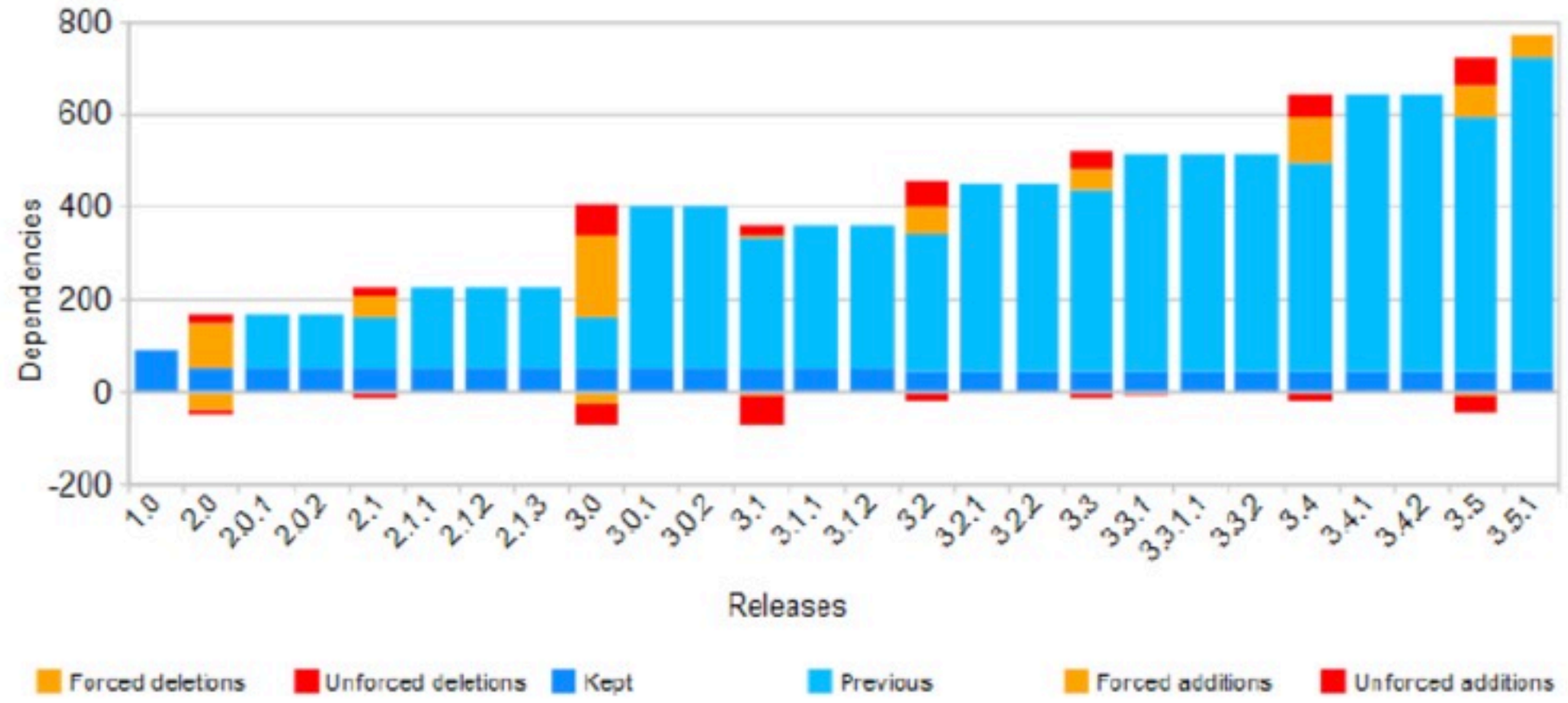


Fig. 5 Evolution of the overall complexity

Michel Wermelinger, Yijun Yu, Angela Lozano, Andrea Capiluppi: Assessing architectural evolution: a case study. Empirical Software Engineering 16(5): 623-666 (2011)

challenges

Ecosystems beyond software

Smart city

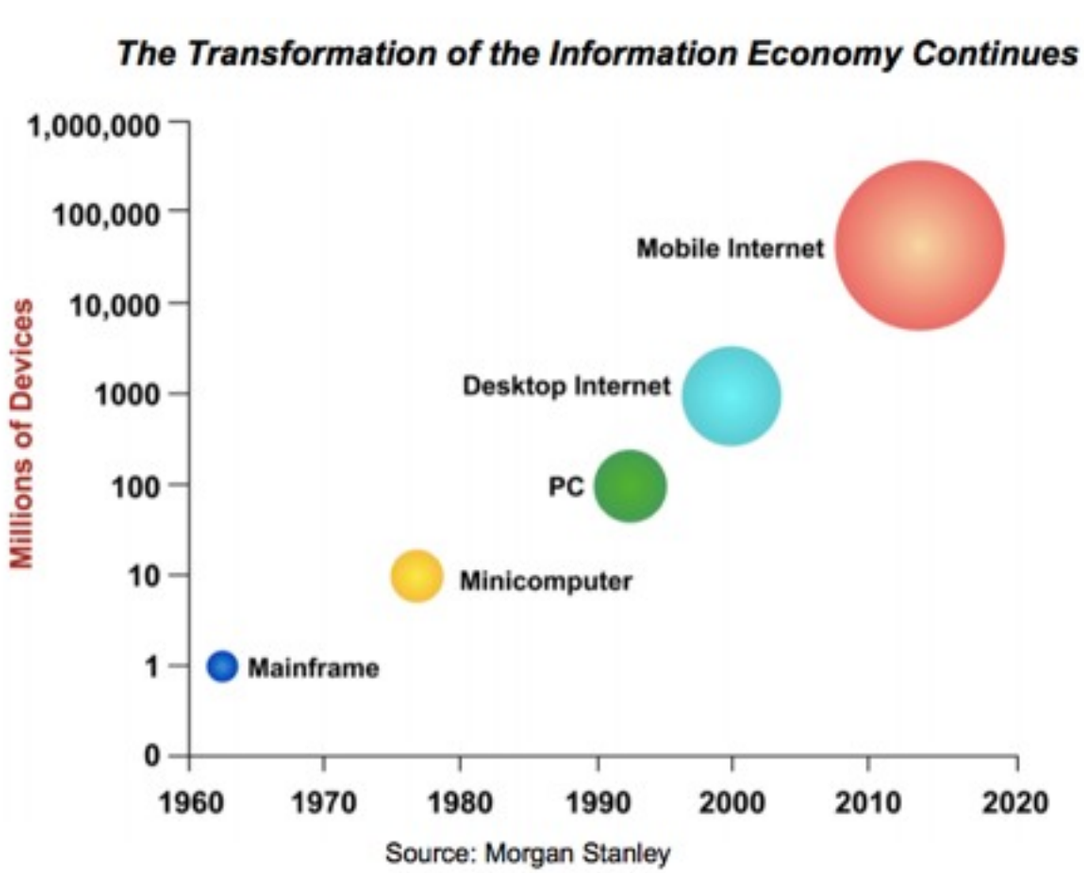
Smart devices

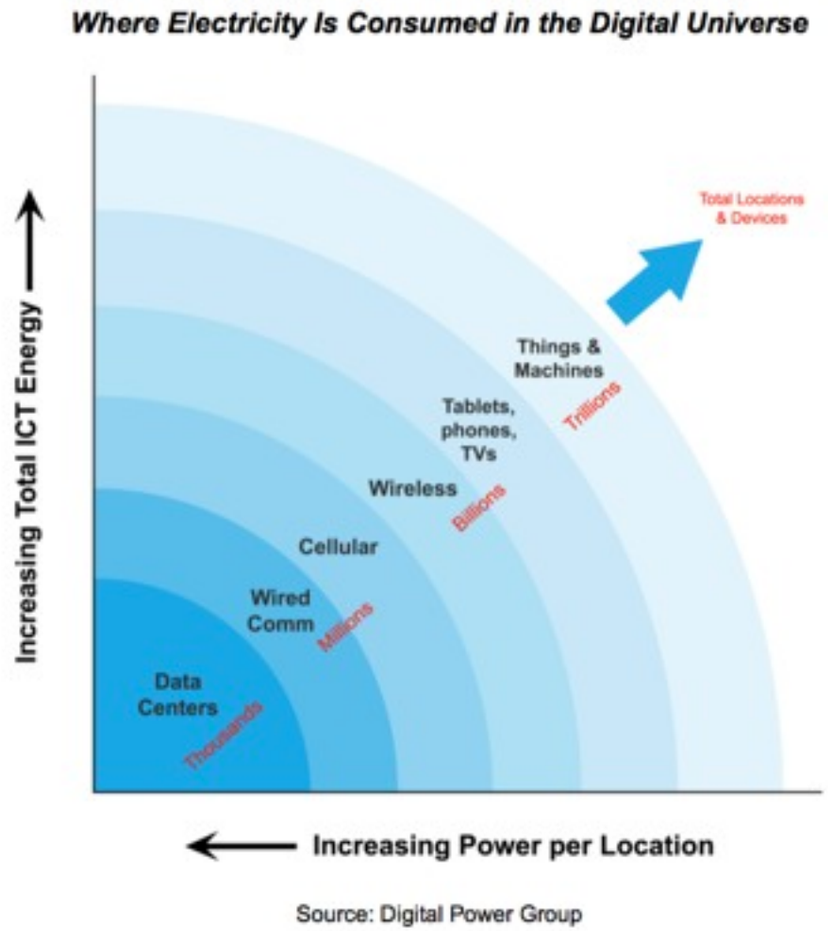
Big data

Data centres

challenges

1 iPhone > 1 refrigerator

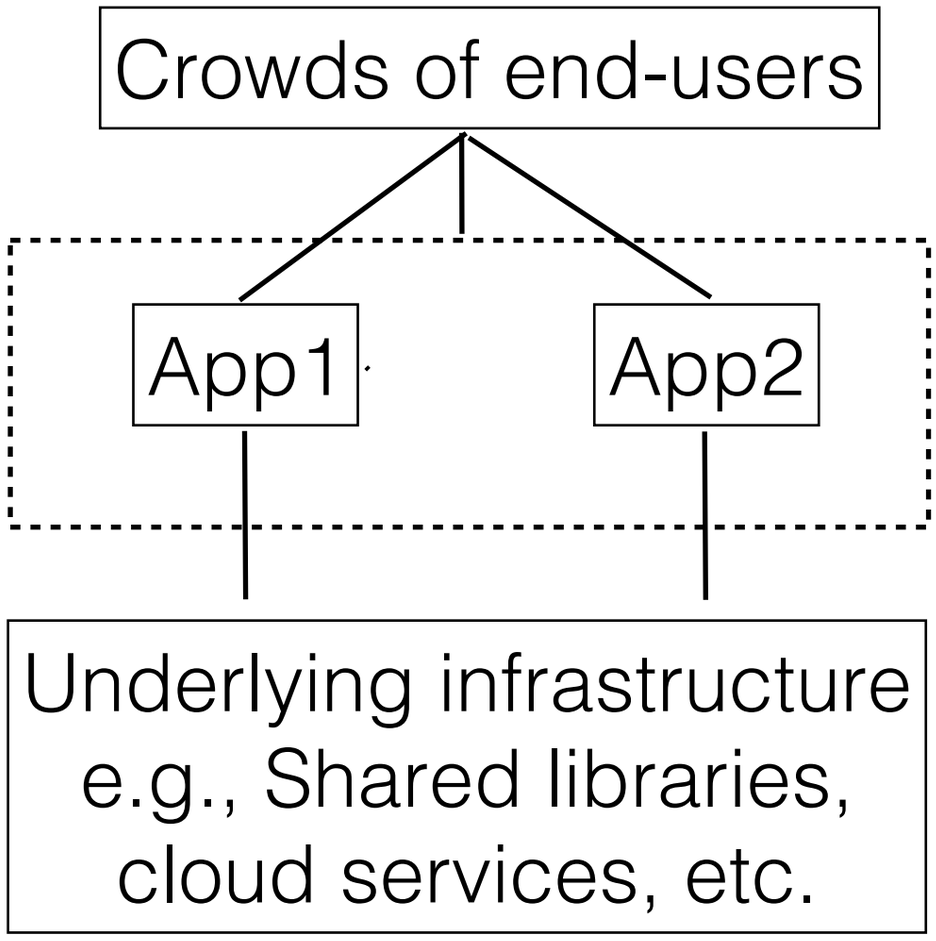




Mark Mills. “An Overview of the Electricity Used by the Global Digital Ecosystem”, Digital Power Group, August 2013.

We need to think out of the box.

Jackson's context diagram and system boundary



example problem

The gcc crossroads

top design goals:

performance

...

energy efficiency

...

PCs

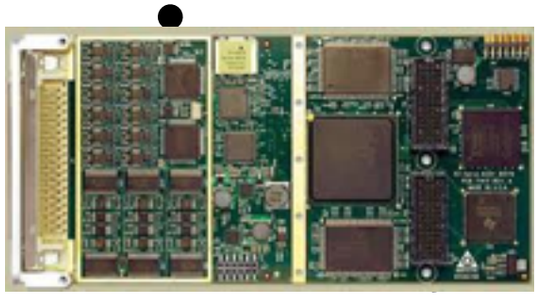
laptops

Servers

Smart phones

IoT devices

Raspberry Pi

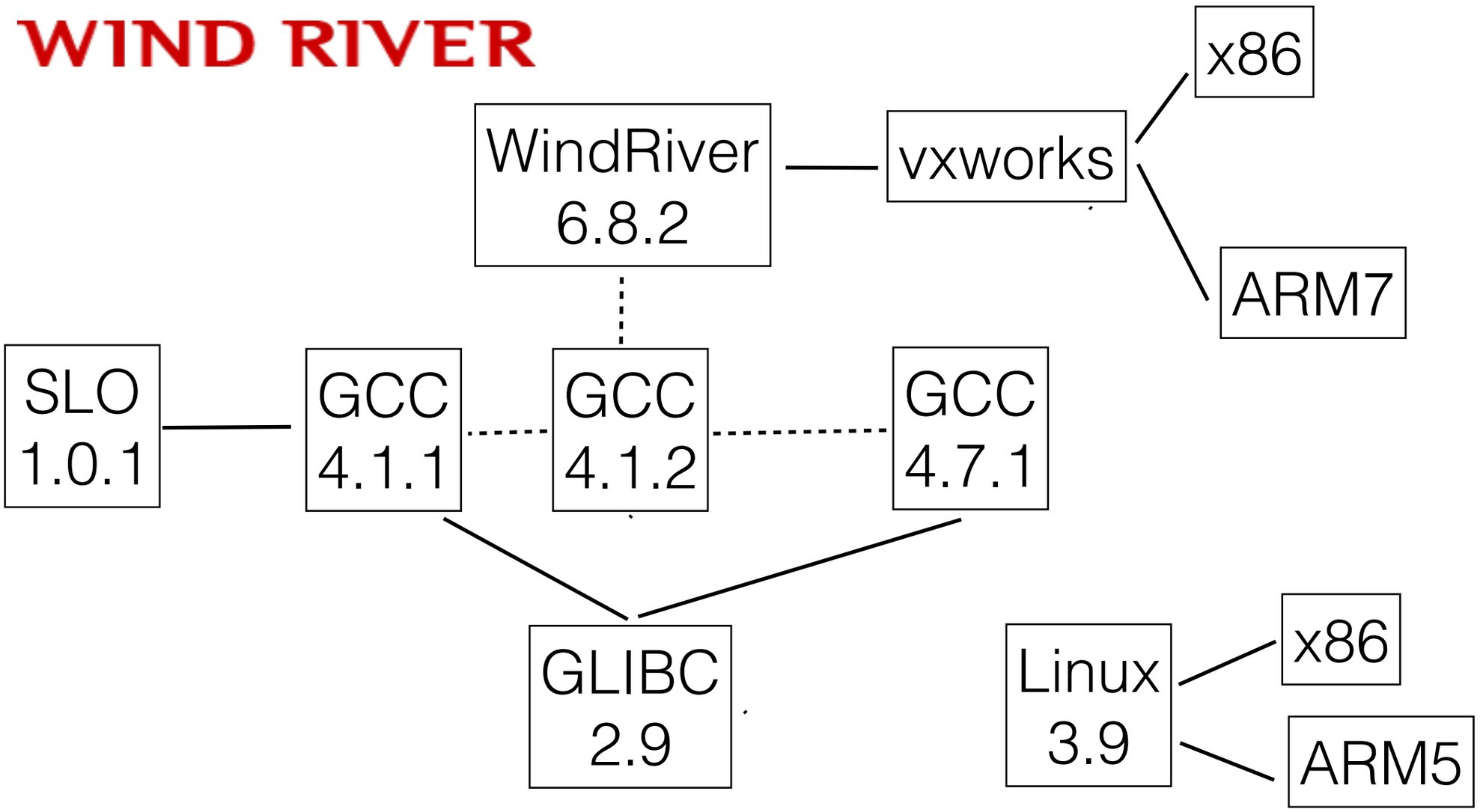


DSP/embedded systems

ebook
readers

example problem

Currently...

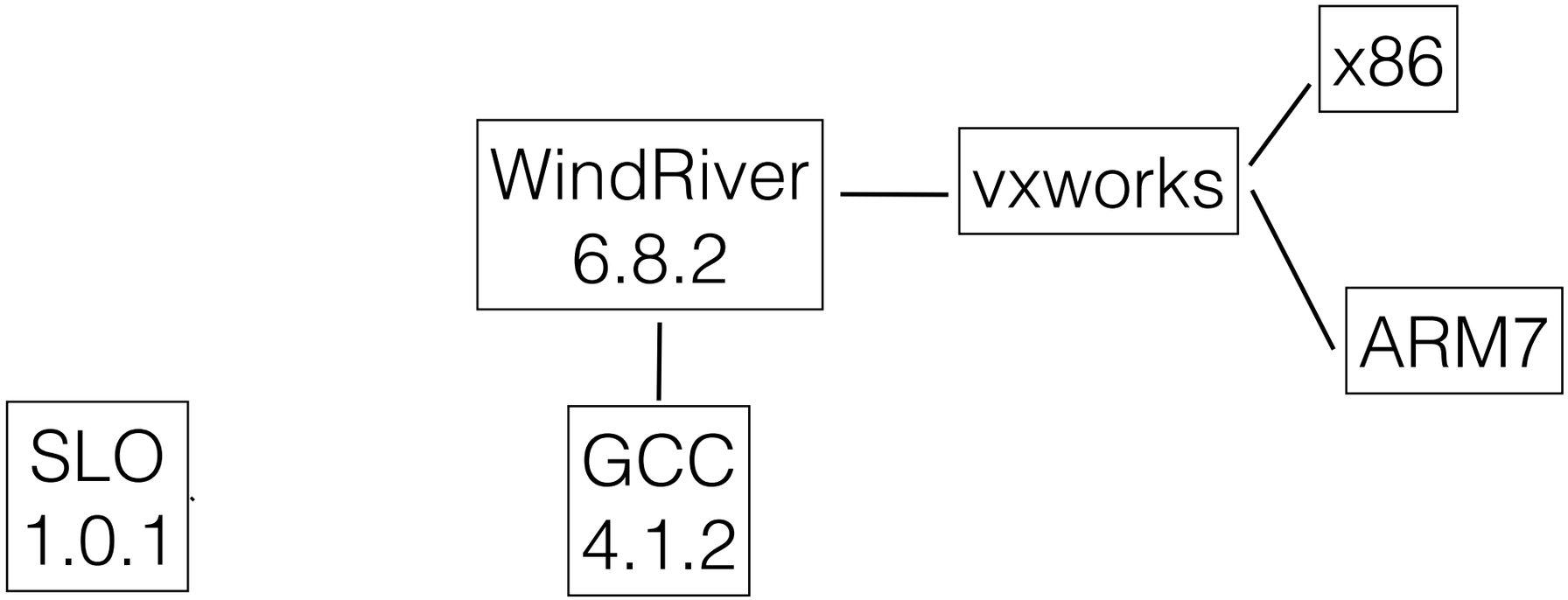


Existing relationship details known to engineer

Relationship known, but details unknown to engineer

example problem

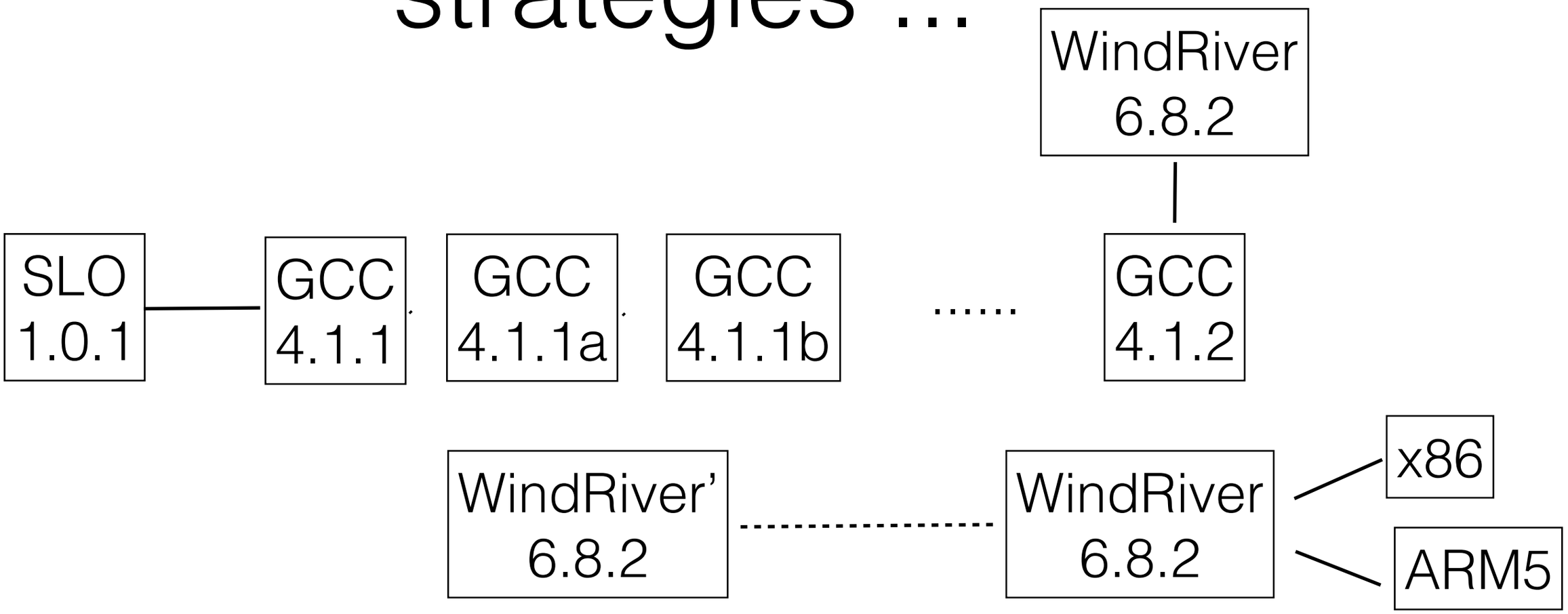
Context of the engineering goal



- Goal: Use SLO 1.0.1 to assess the performance of embedded system programs running on the VxWorks behind WindRiver 6.8.2
- Size of the embedded system ~10M LOC

example problem

Adaptation (manual switching) strategies …

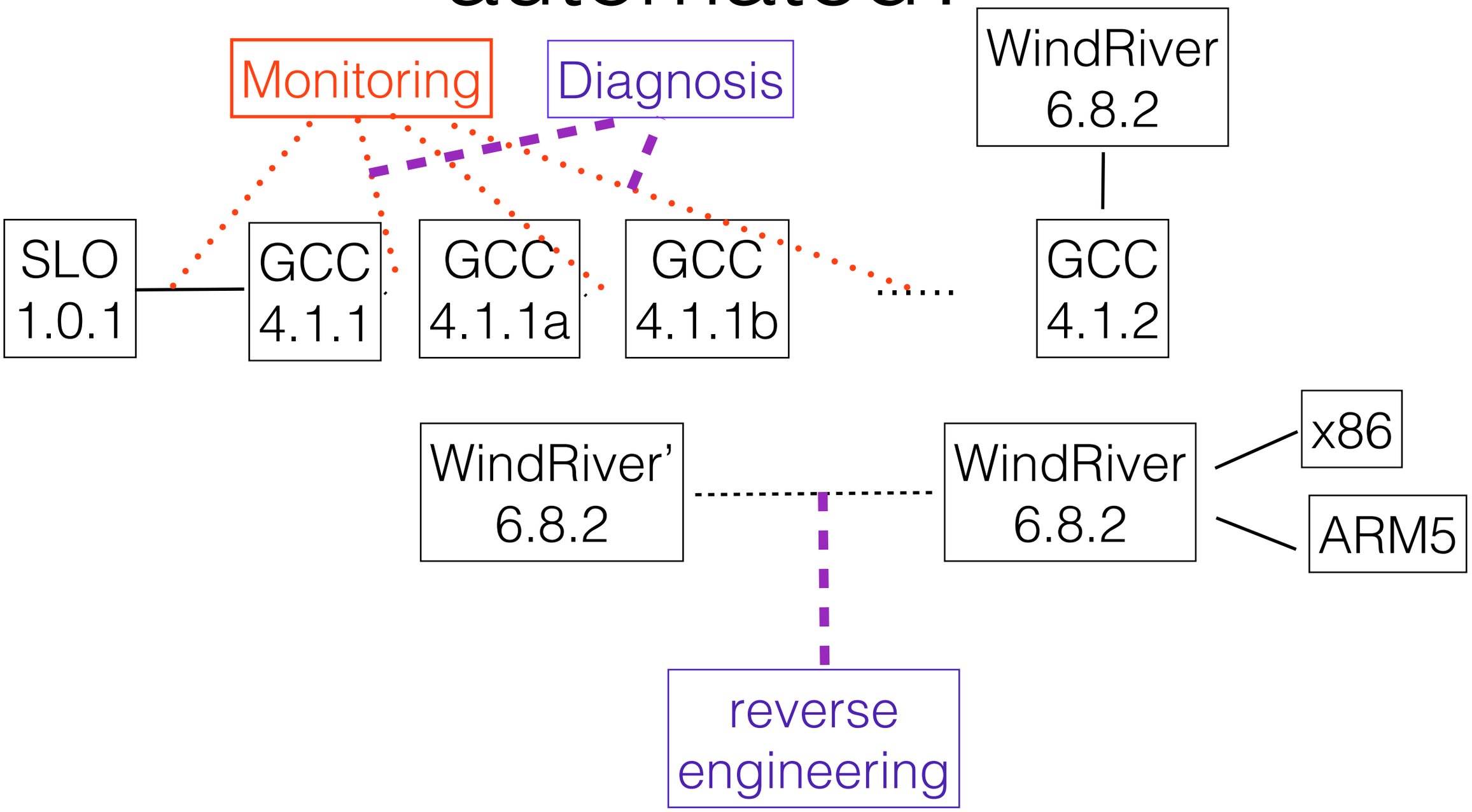


- For every revision in GCC between 4.1.1 and 4.1.2, find out whether SLO can work with GCC or not?

- For every compiler option in Linux and VxWorks configurations, check what changes are necessary to make the system build results compatible?

example problem

What can make the strategies automated?



heuristics from the example problem

How to automate these strategies?

- **Monitoring/Diagnosis:** design diagnosis tasks from meaningful change logs

Yiqiao Wang, Sheila A. McIlraith, Yijun Yu, John Mylopoulos: **Monitoring and diagnosing software requirements**. Autom. Softw. Eng. 16(1): 3-35 (2009)

Yijun Yu, Thein Than Tun, Bashar Nuseibeh: **Specifying and detecting meaningful changes in programs**. ASE 2011: 273-282

- **Monitoring/Switching:** decide suitable monitoring conditions

Mohammed Salifu, Yijun Yu, Arosha K. Bandara, Bashar Nuseibeh: Analysing monitoring and switching problems for adaptive systems. Journal of Systems and Software 85(12): 2829-2839 (2012)

- **Invariant traceability:** reduce change propagations to save efforts

Yijun Yu, Yu Lin, Zhenjiang Hu, Soichiro Hidaka, Hiroyuki Kato, Lionel Montrieux: **Maintaining invariant traceability through bidirectional transformations**. ICSE 2012: 540-550

- **Reverse engineering:** learn behaviour models from code patches

Yijun Yu, Yiqiao Wang, John Mylopoulos, Sotirios Liaskos, Alexei Lapouchnian, Julio Cesar Sampaio do Prado Leite: **Reverse Engineering Goal Models from Legacy Code.** RE 2005: 363-372

FAQ about Time Travel

Food of thoughts

- Does capability of predictive analytics change the rules of an adaptive time machine?
  - Paradox 1: you can't kill your ancestors
  - Paradox 2: you cannot meet yourself in future
- Side-effects of predicative analytics must be controlled
  - feedback loop needs to be sustainable