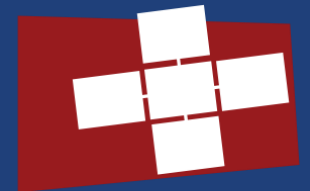




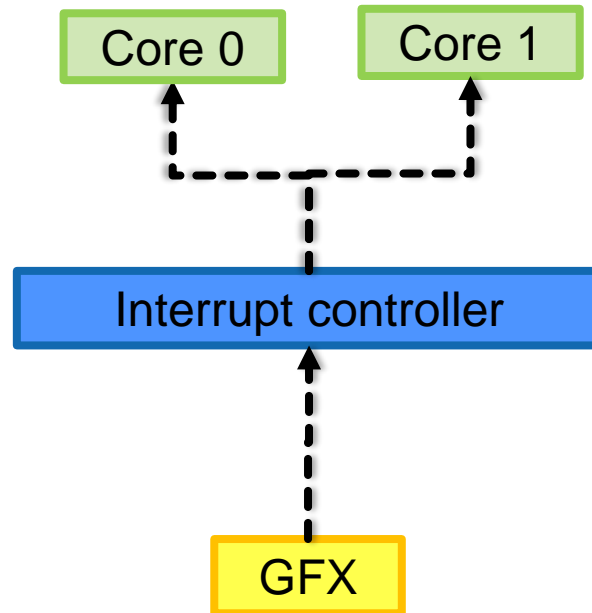
Formalizing Interrupt Systems

11th EuroSys Doctoral Workshop – Lukas Humbel

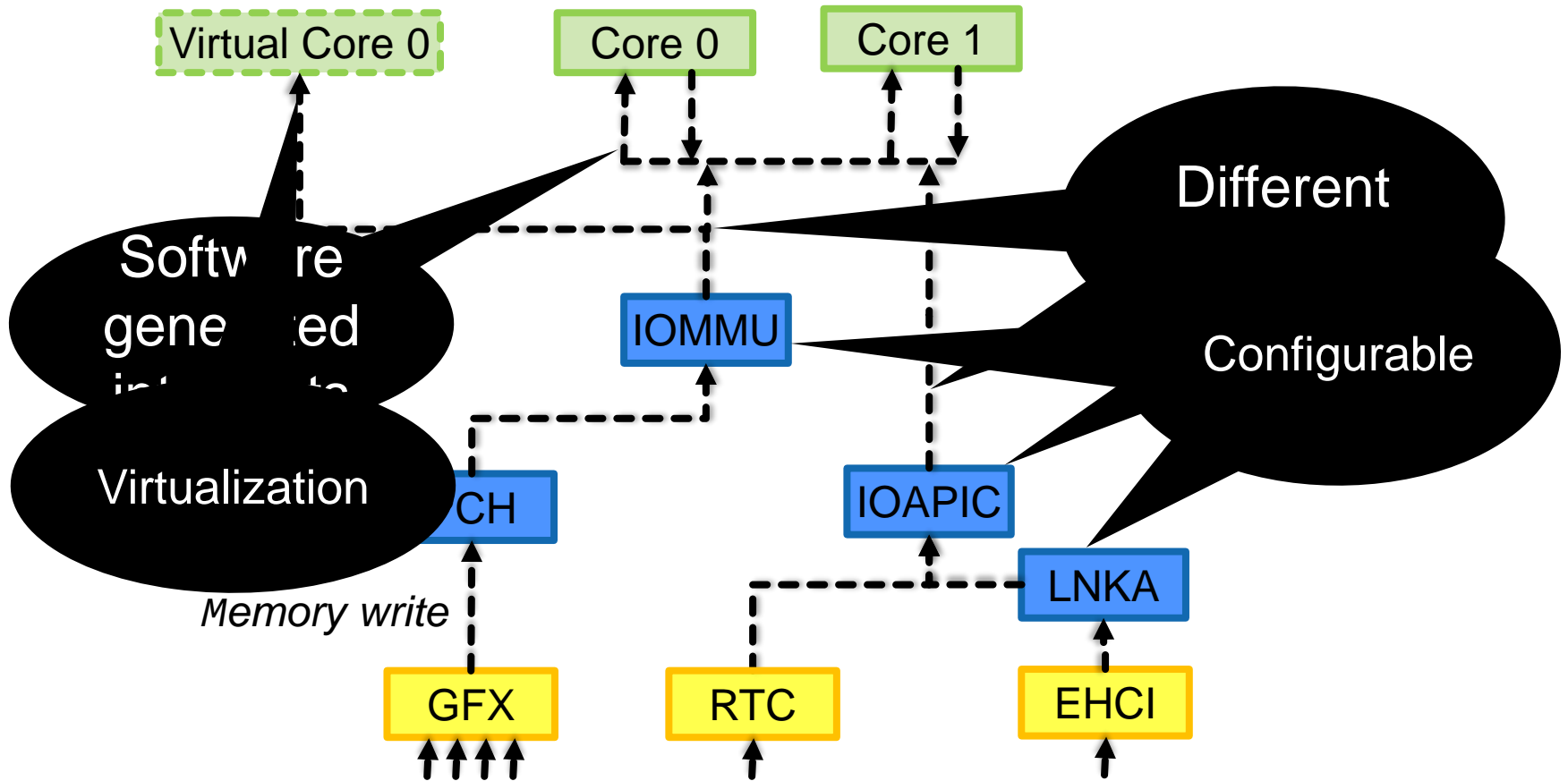


Systems@ETH zürich

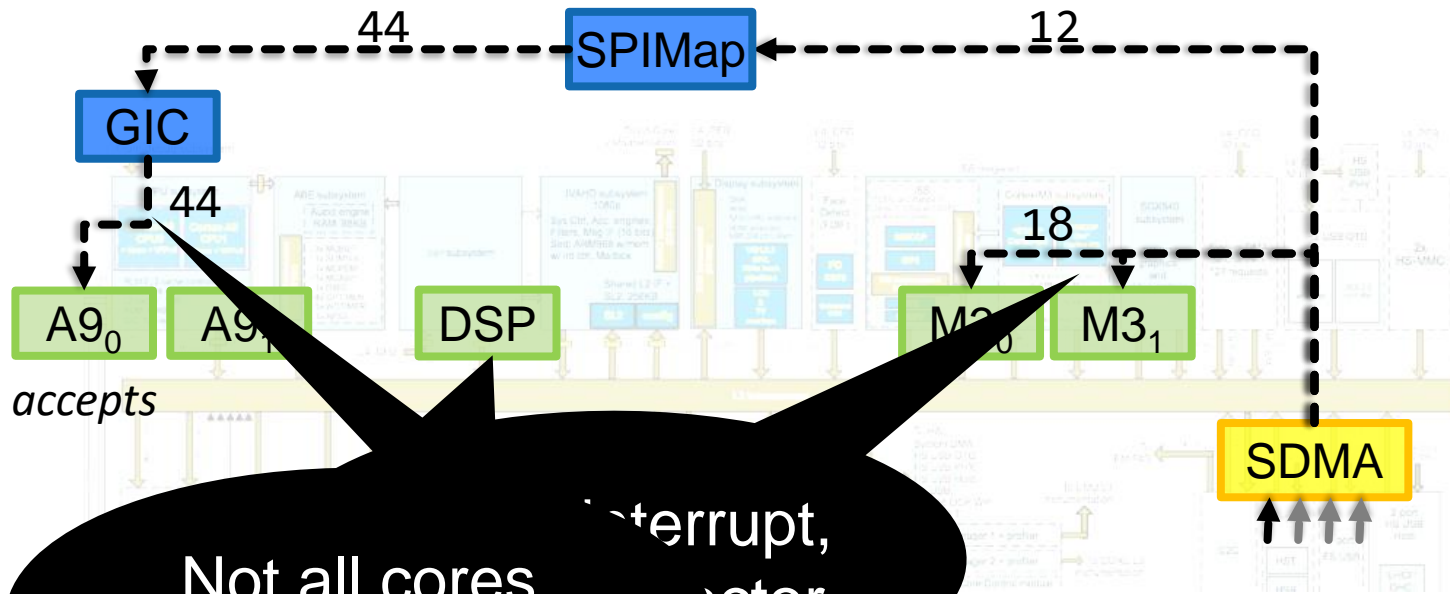
The Ideal System



A Real Intel System



A Real ARM SoC



Not all cores are reachable by interrupt, vector

Trends

Hardware **complexity**
and **diversity**.

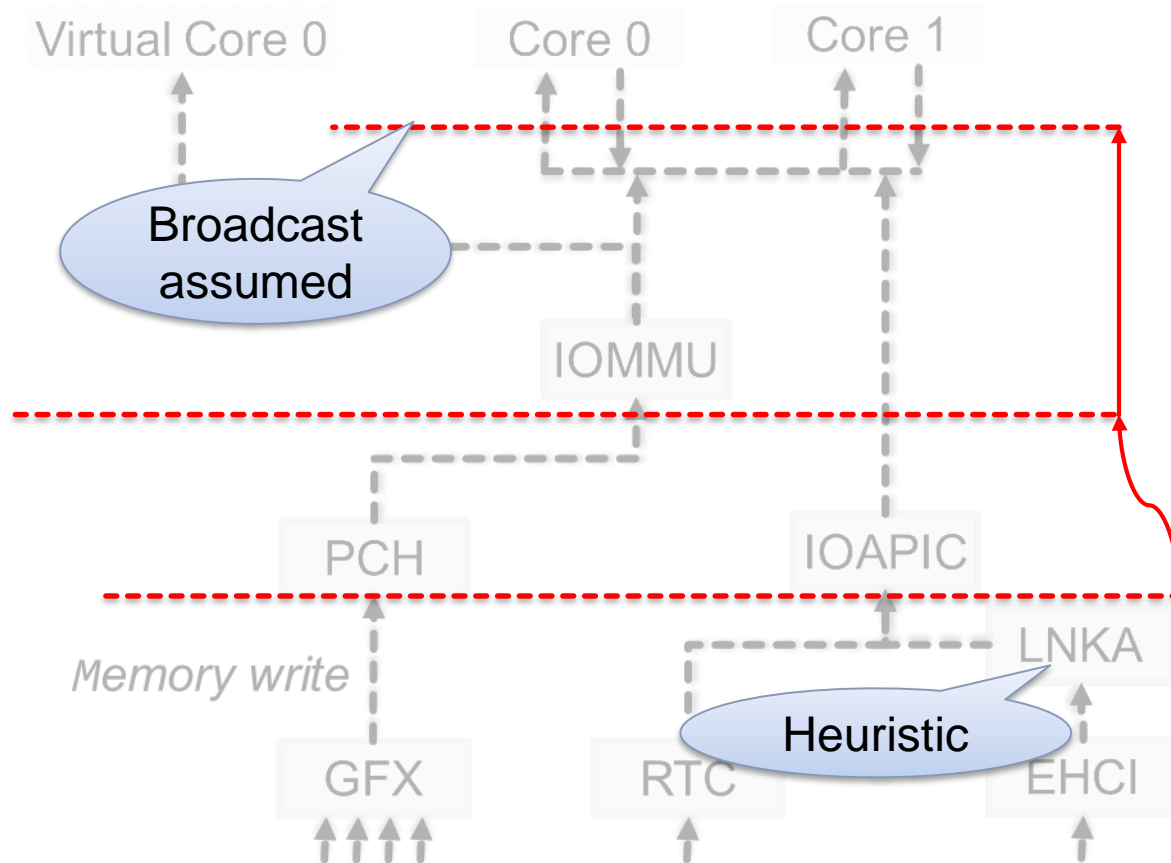
No **formal**
description of
hardware.

New **demands**:

- Direct-device access
- CPU hot-plug

→ A lot of C code that
might be correct
→ Inflexible OS

Representation - Linux

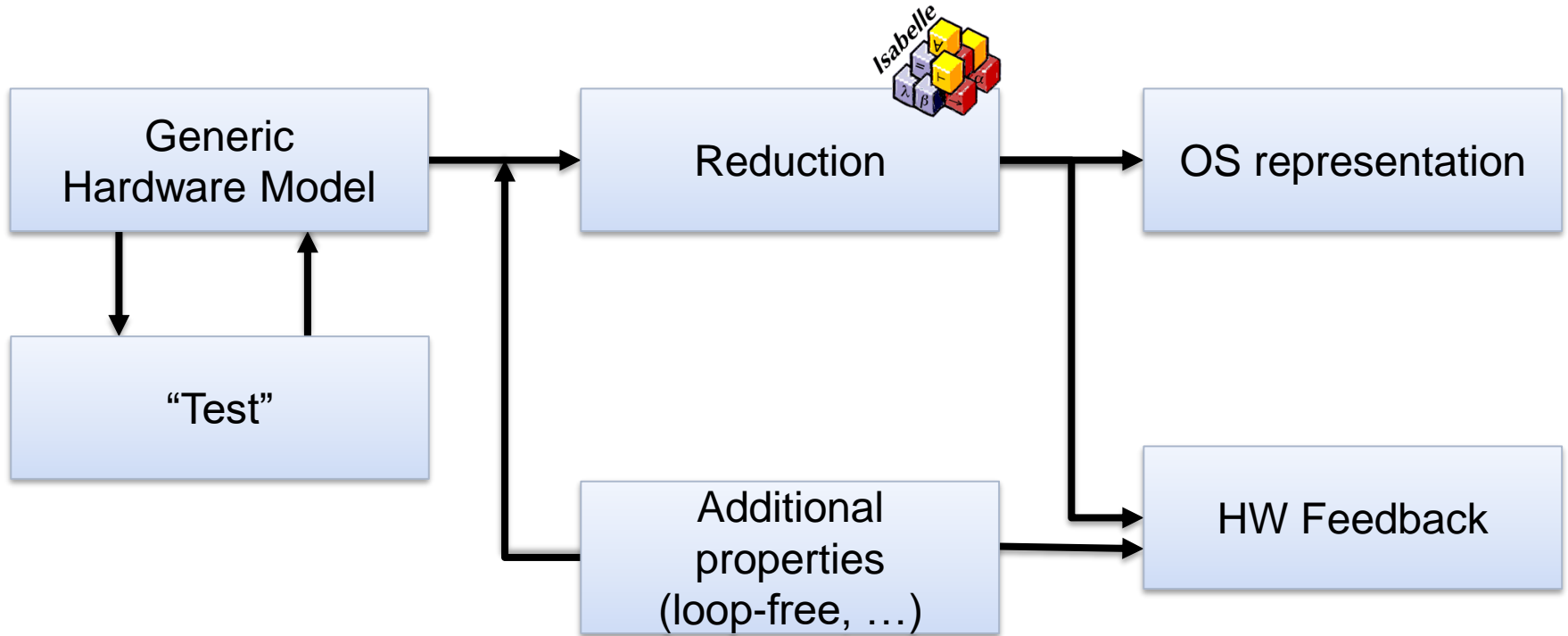


Hard to handle heterogeneity (M3 cores)

Problem

- How to write **generic** and **correct** interrupt configuration code?
- What is good interrupt hardware?

Our approach



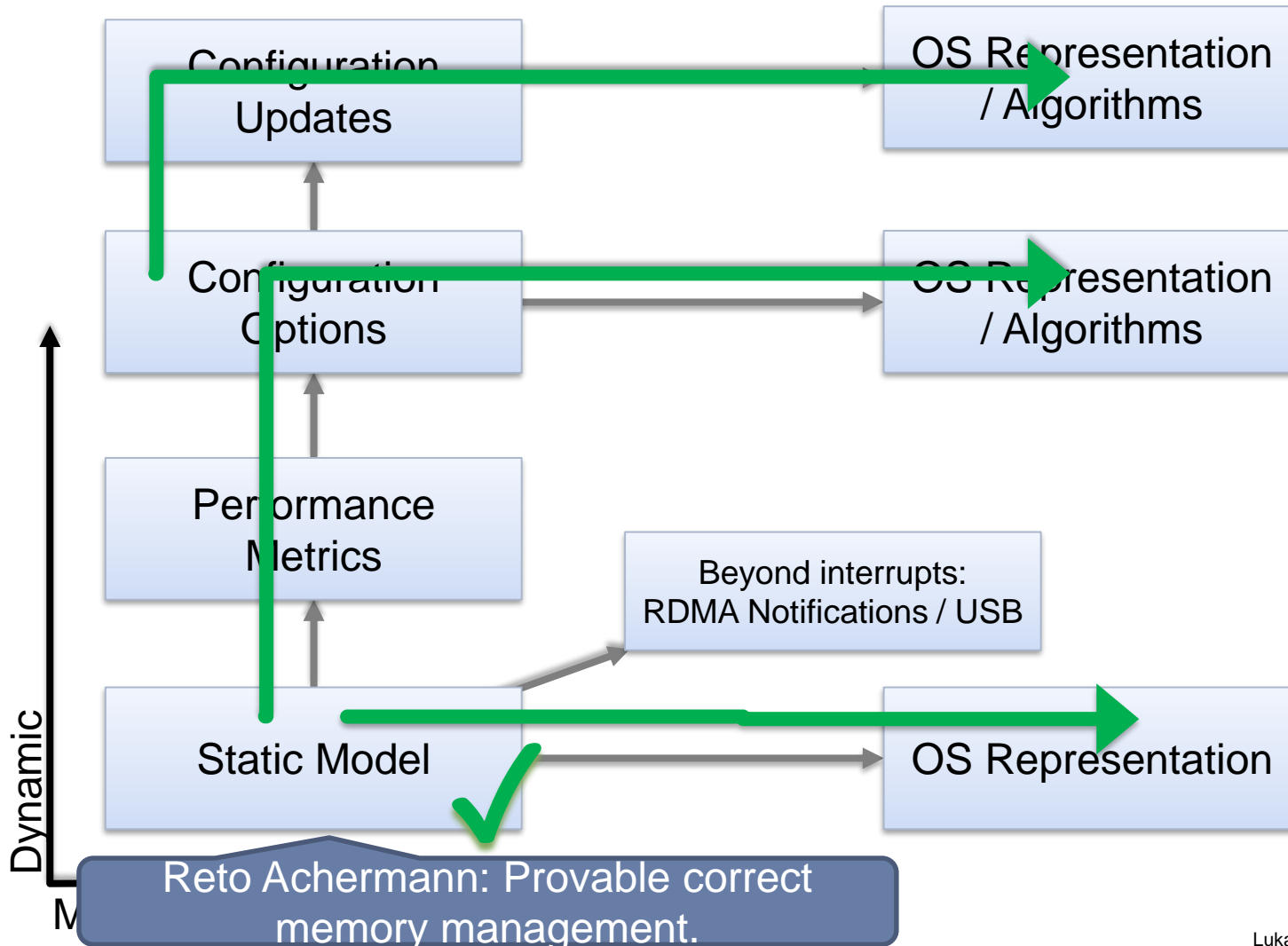
Improve OS Design by:

- Being clear what HW properties are assumed
- Separating concerns
- Providing basis for verification

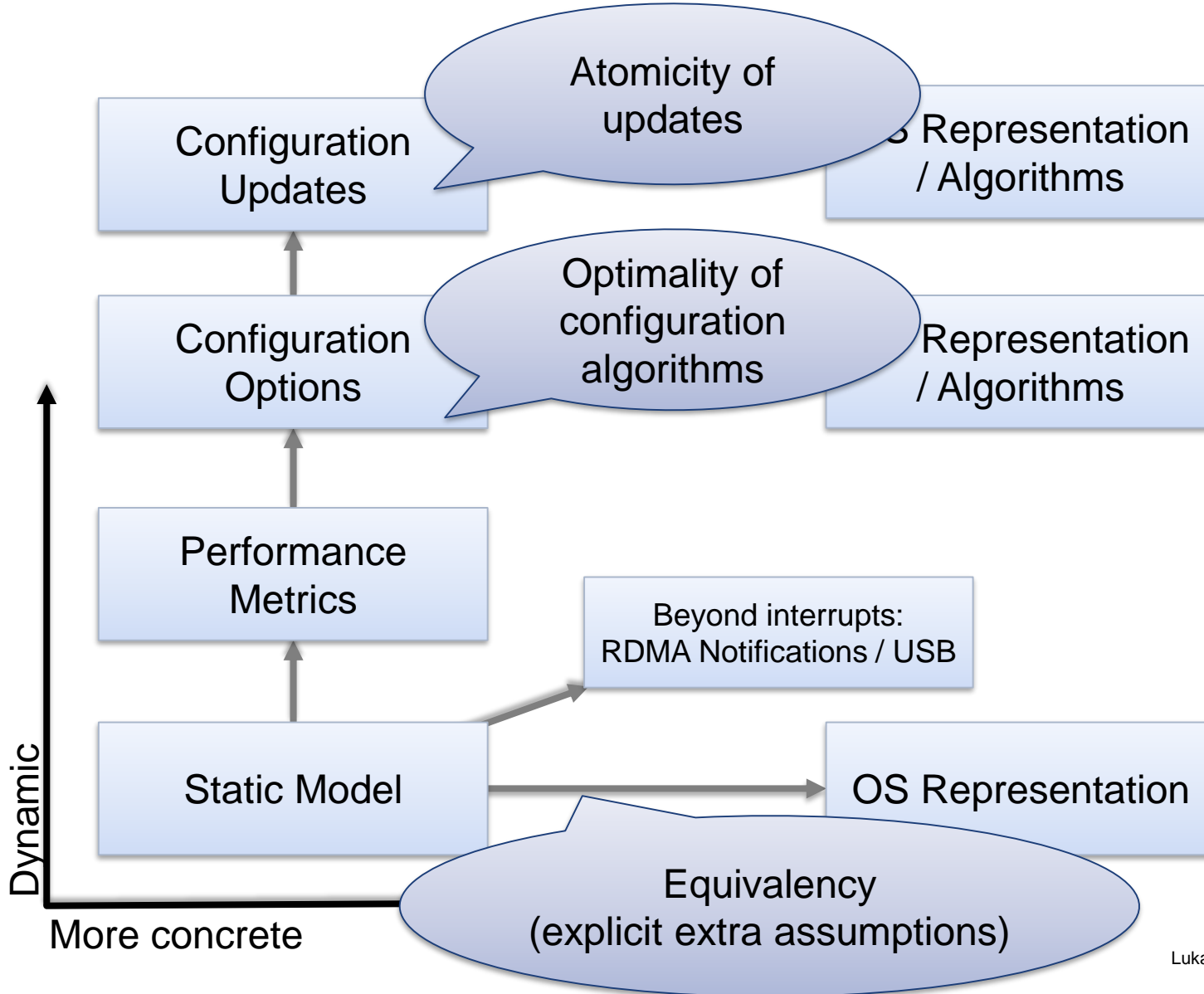
Improve HW:

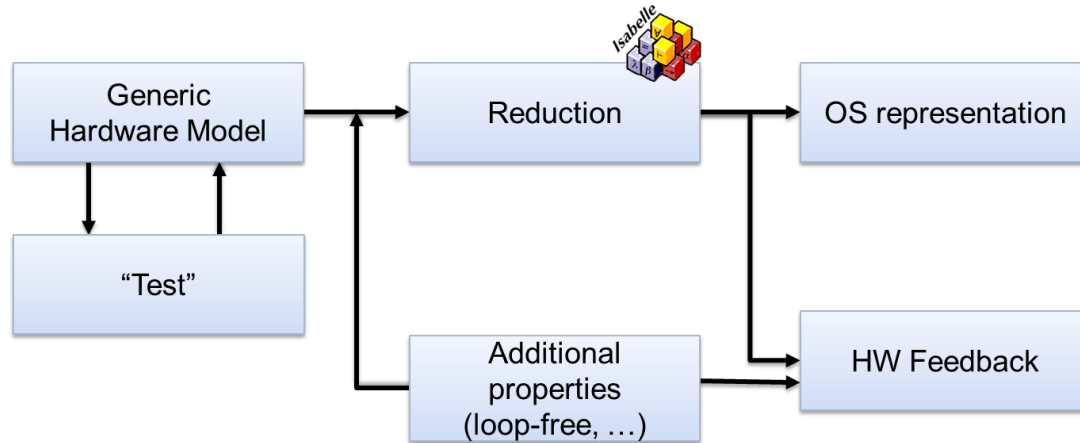
- By deriving what properties lead to simpler abstractions.

Overview



Overview – Formal methods





Thanks for listening, Questions?