Capabilities in Barrelfish

Mothy, Barrelfish crew, ETH Zurich

Channelling: David Cock, Simon Gerber, Gerd Zellweger, Reto Achermann, Lukas Humbel, and others

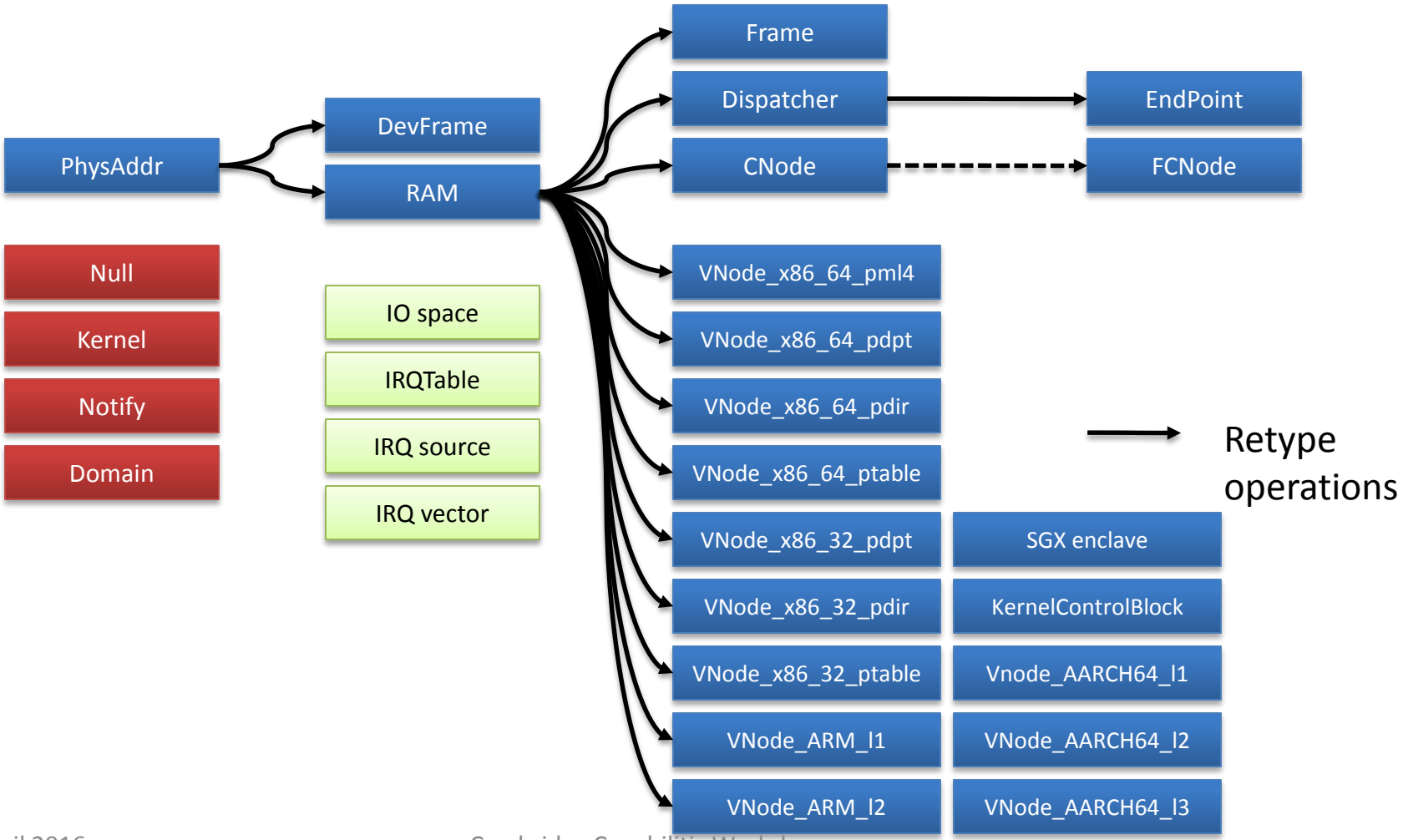
Delta from seL4: lots of types

Systems@ETH zürich

- Many simultaneous architectures
  - Types for all hardware structures
- Hamlet:
  - DSL for capability type system
  - Fields, formats, semantics, retype rules
  - Generate dispatch/invoke/retype code

Some of the Barrelfish capability type system

Systems@ETH zürich



Delta from seL4: physical address spaces

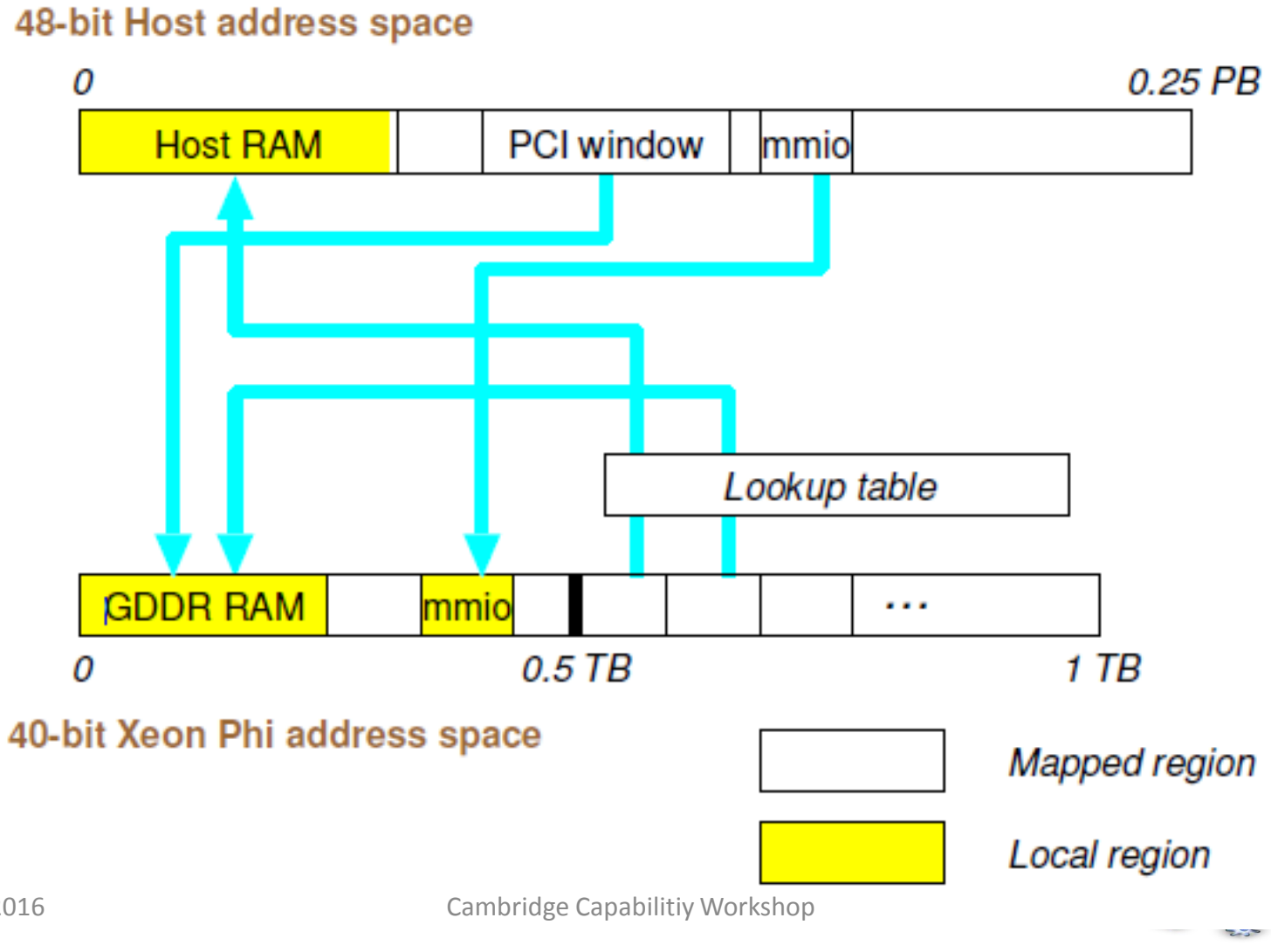
Systems@ETH zürich

- Capabilities relative to some physical address space
  - Lots of these: 32-bit identifier
  - At least one per core, and usually more

⇒ Global ids even for resources that only accessible locally.

- Makes sense for modern hardware:
  - SoC / NoC messes
  - Remote buffer allocation at rackscale
  - Core sequestering in large machines
- Capability resolution process
  - To local address space

Simple example: Xeon Phi

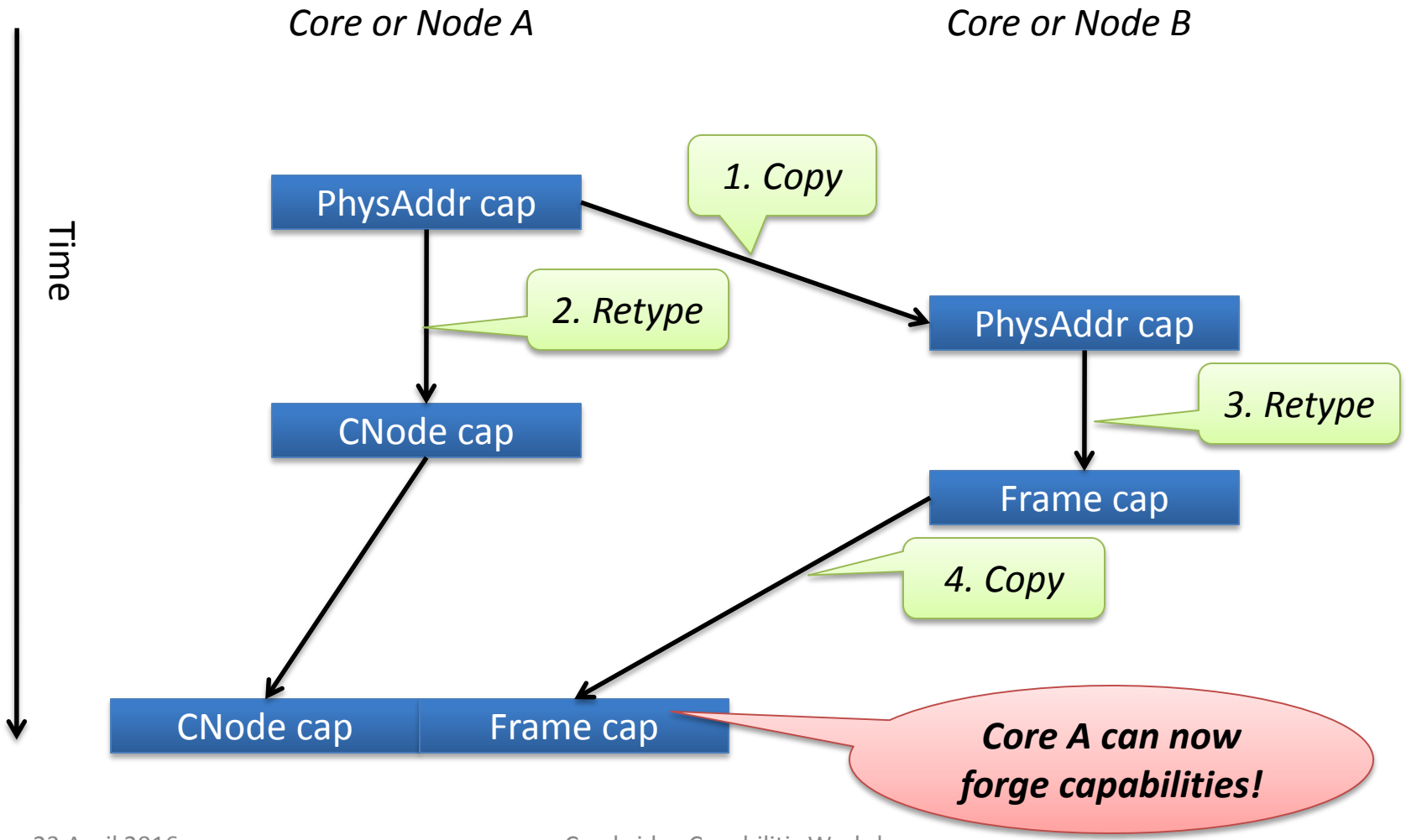


Barrelfish is distributed

Systems@ETH zürich

- Multiple capability databases
  - Coexisting in the same physical address space
- Capabilities must be copied between cores
  - May require serialization
  - Over some (not all) message channels
- Implications:
  - Need to identify all local copies of a capability based on bits, not reference.
    ⇒ Capabilities must be indexed

Retyping requires consensus



CHERI and Barrelfish

Systems@ETH zürich

- Format flexibility is great
  - Easy to represent BF capabilities as CHERI ones
  - Barrelfish mapping caps ⇔ CHERI virtual caps?
- What privilege is required to retype?
- Big challenges:
  - Serialization
  - Revoke / retype
    - Quickly find all copies / derivations