Deconstructing MinBFT for Security and Verifiability

Vincent Rahli,
Francisco Rocha,
Marcus Völp, and
Paulo Esteves-Verissimo

http://wwwen.uni.lu/snt/research/critix

March 15, 2016

Meet The Team

securityandtrust.lu

CritiX

Francisco Rocha

Marcus Völp

Jérémie Decouchant

Paulo Esteves-Verissimo

Vincent Rahli

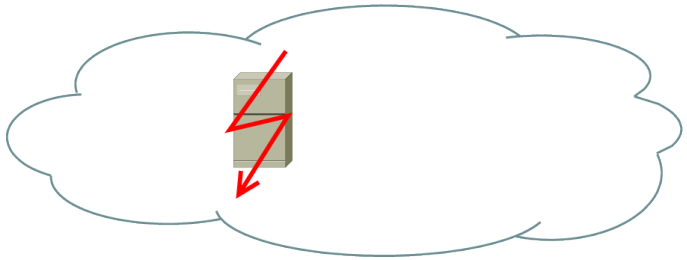
uni.lu UNIVERSITÉ DU LUXEMBOURG

Overall Story

securityandtrust.lu

CritiX

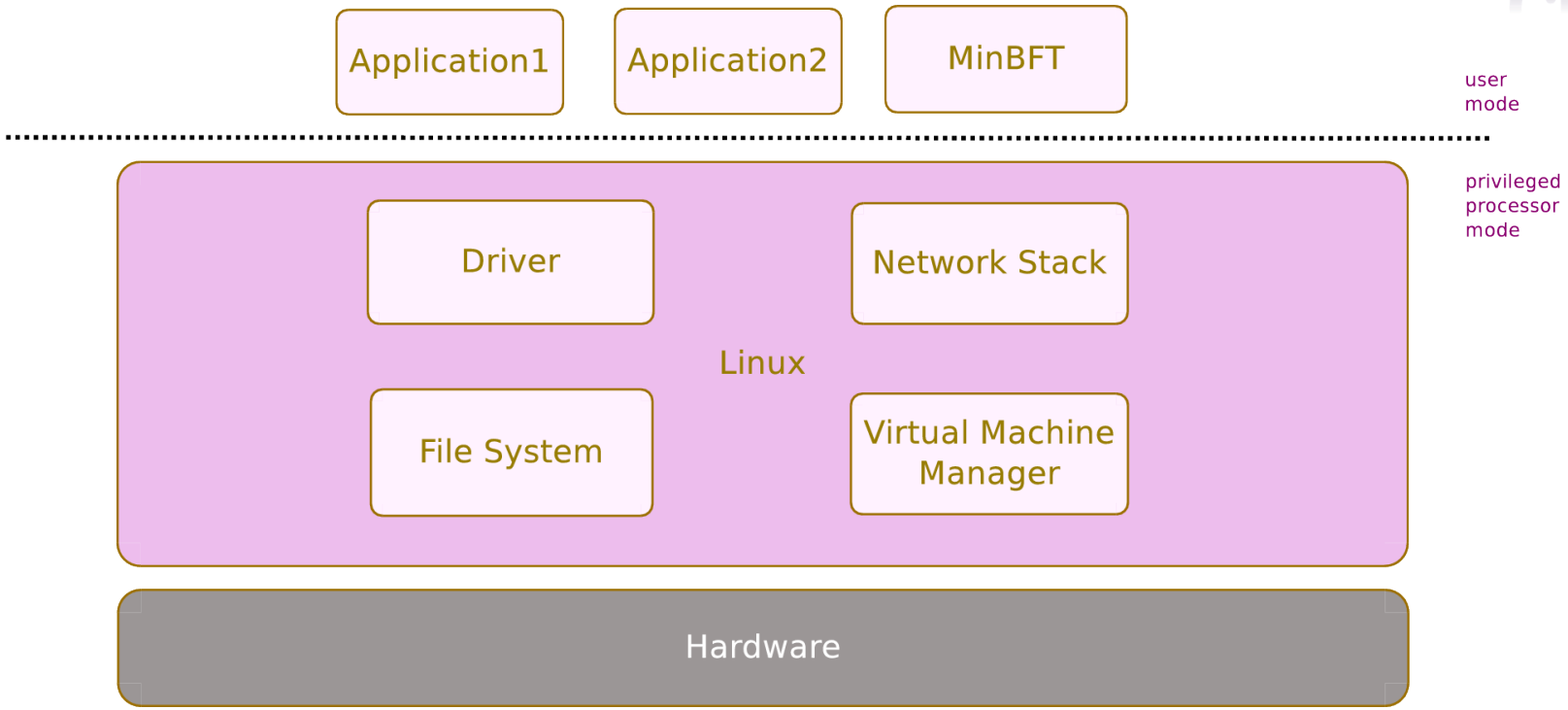
Resilience (fault-tolerance)



Security

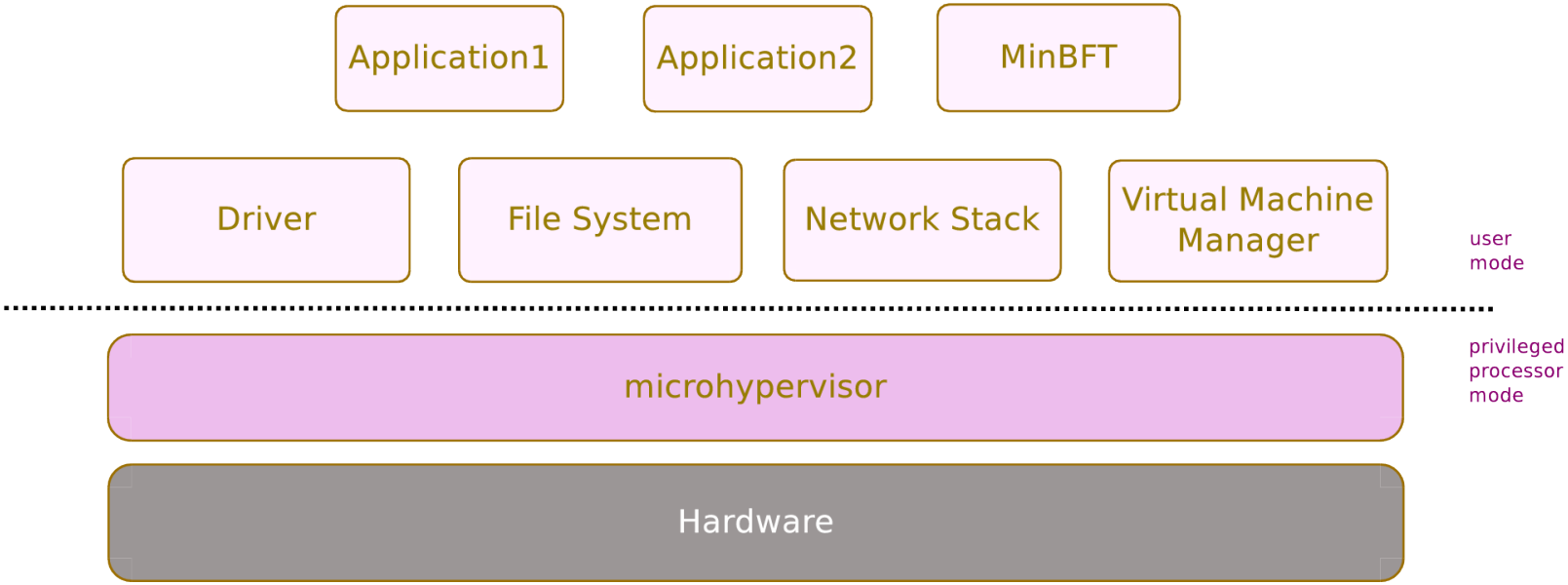
Formal Correctness

Overall Story



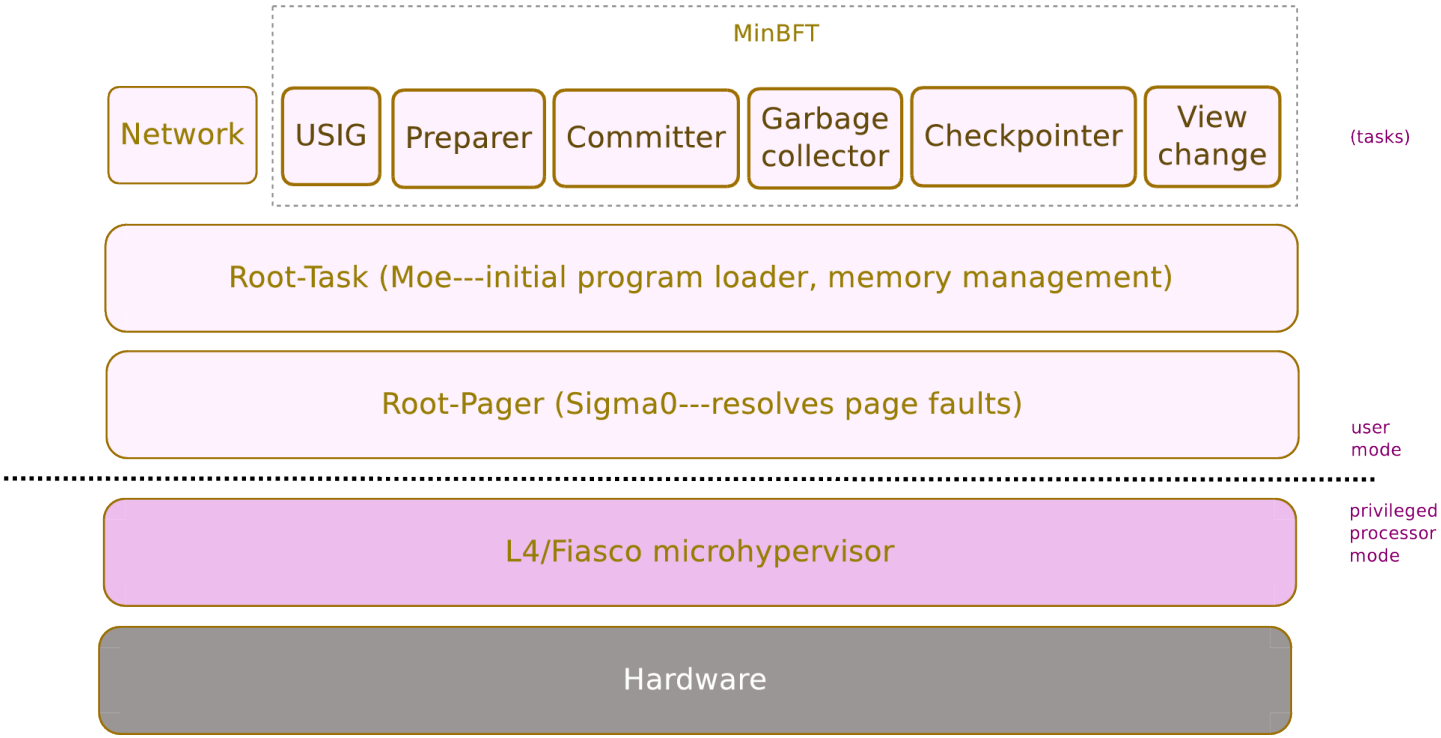
All services in the kernel

Overall Story



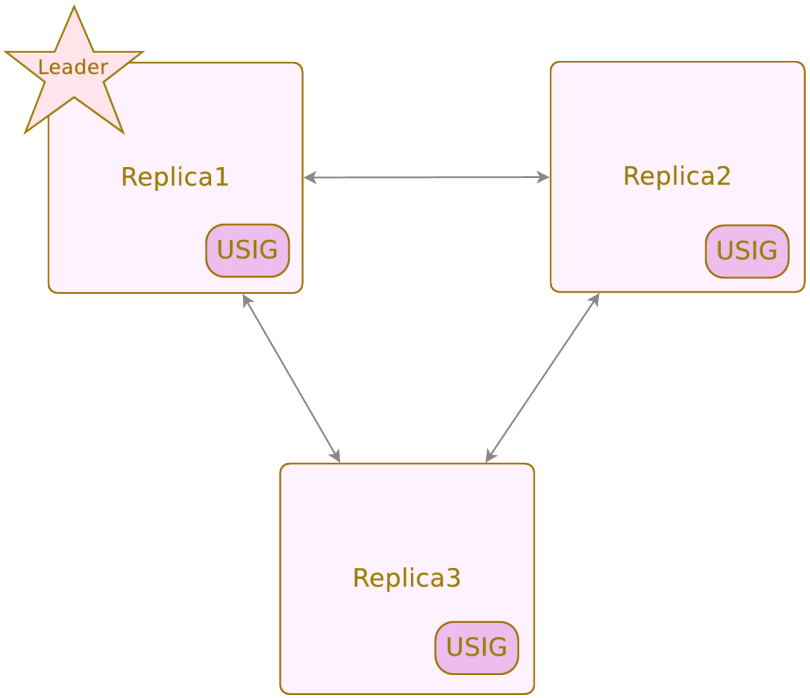
Only keep the bare minimum in the kernel

Overall Story



Build small/secure/verifiable components on top of Fiasco

MinBFT



Byzantine fault-tolerant protocol similar to PBFT

$2f + 1$ as opposed to $3f + 1$ in PBFT

Uses a trusted counter (USIG)

L4/Fiasco

securityandtrust.lu

CritiX

Only program that runs in privileged processor mode

Small: only has what cannot be implemented as the user level

Provides memory isolation

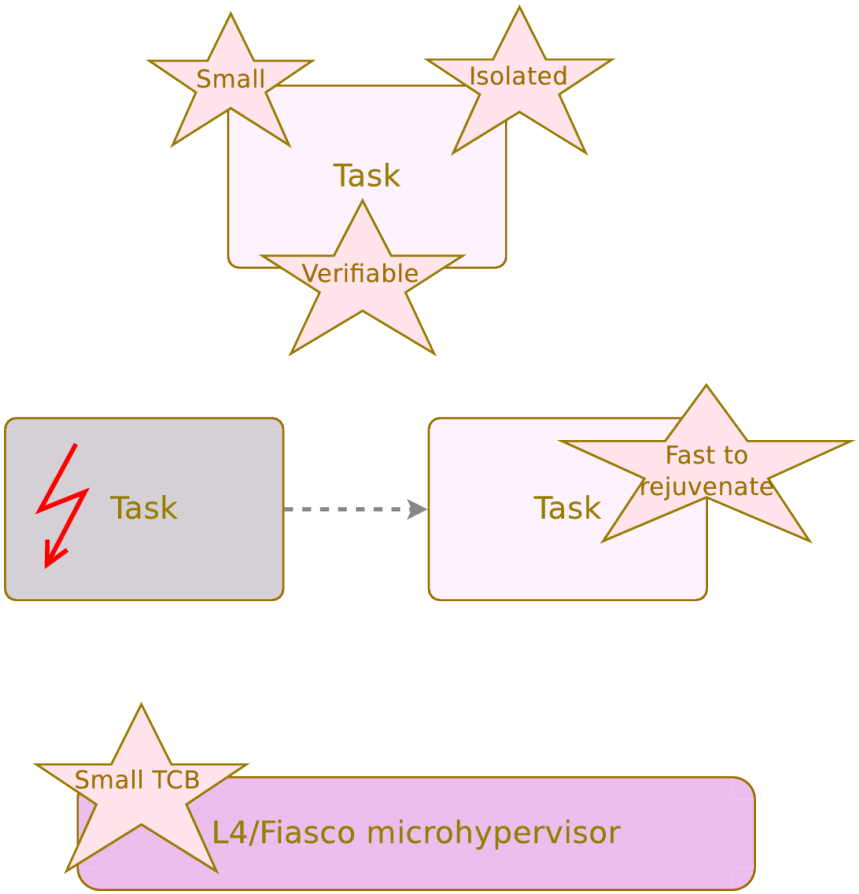
Selective trustworthiness (choose what to use at the user level)

L4/Fiasco

securityandtrust.lu

- Multi-processor support
- System calls using capabilities
- Communication through synchronized IPC calls
- L4RE (Runtime Environment) for application development

Designed for Security

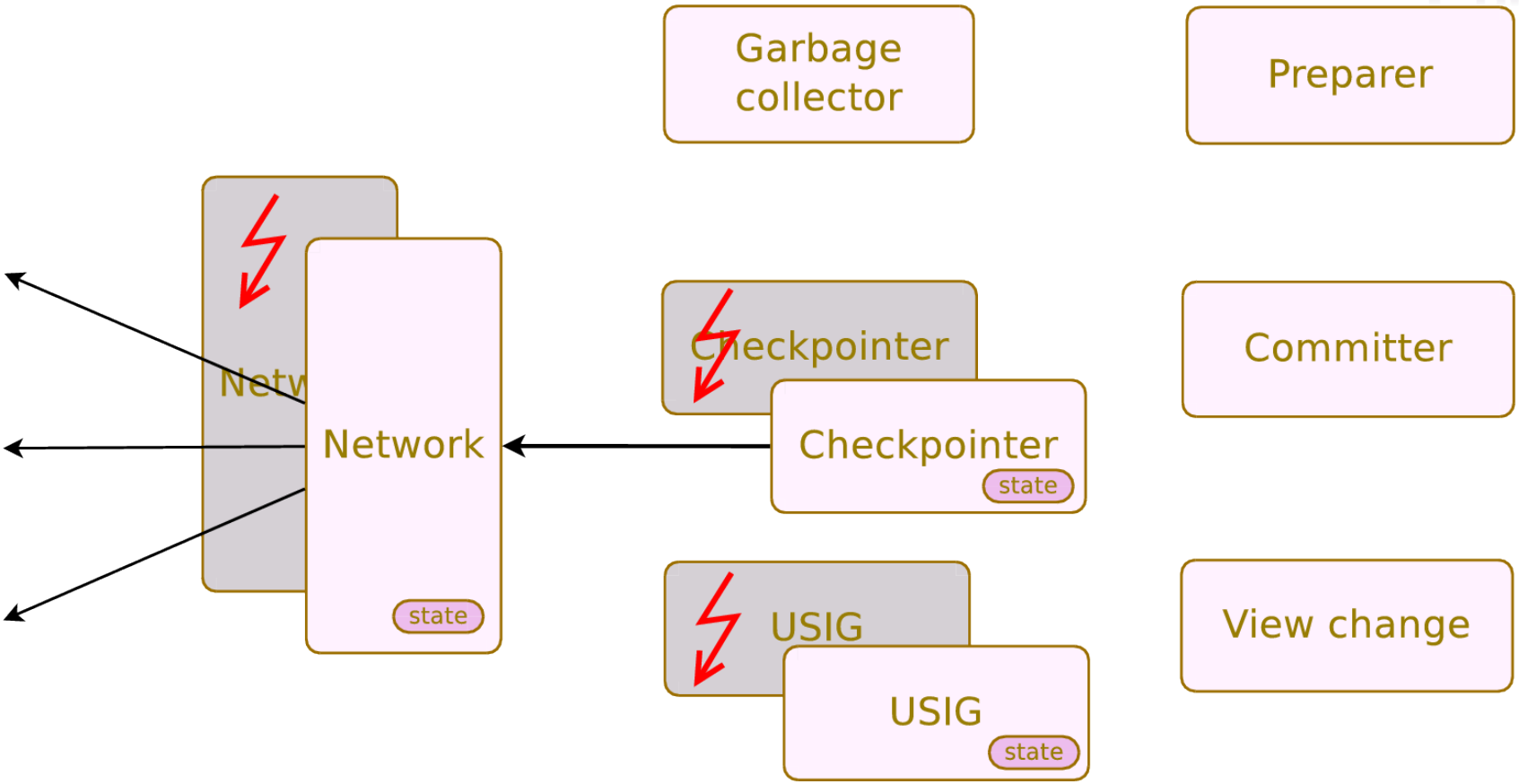


Small isolated verifiable components

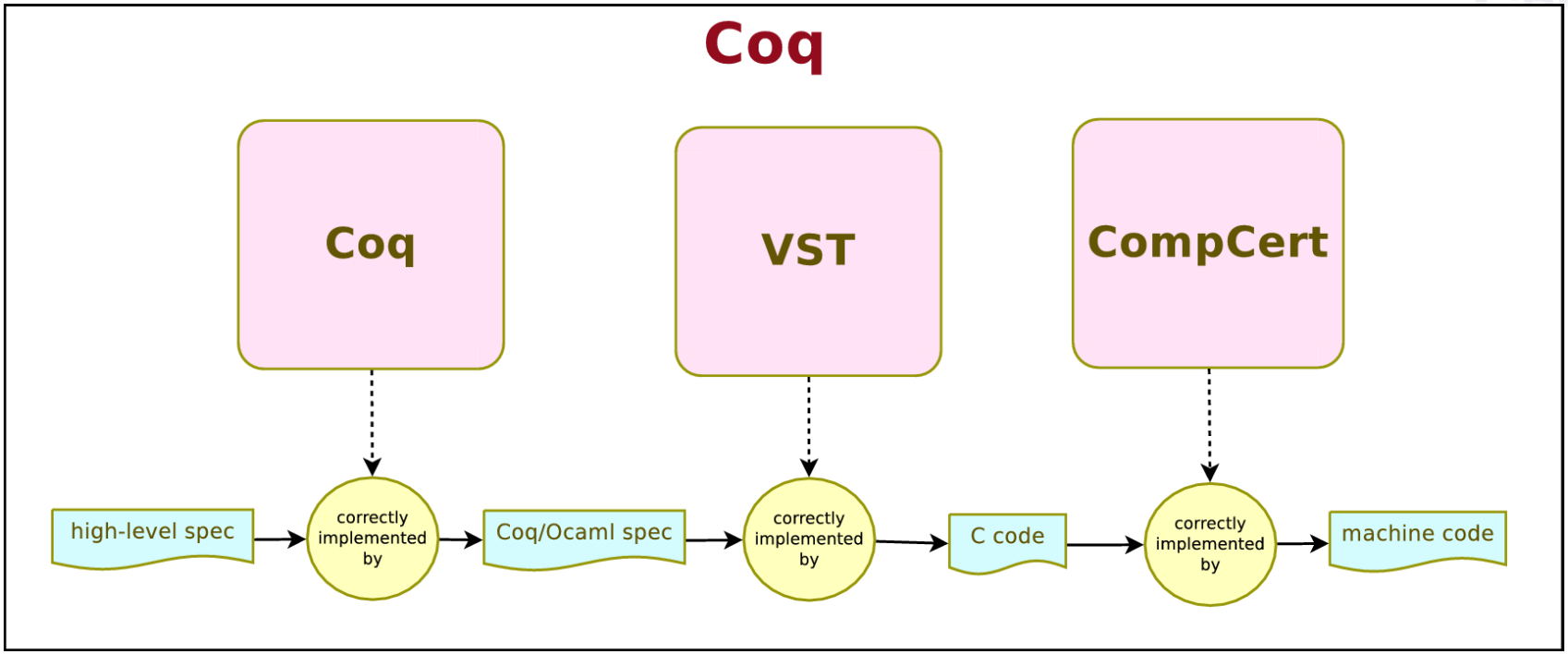
Easier/faster to restart
Selective rejuvenation

Small trusted base:
L4/Fiasco

Fast Recovery



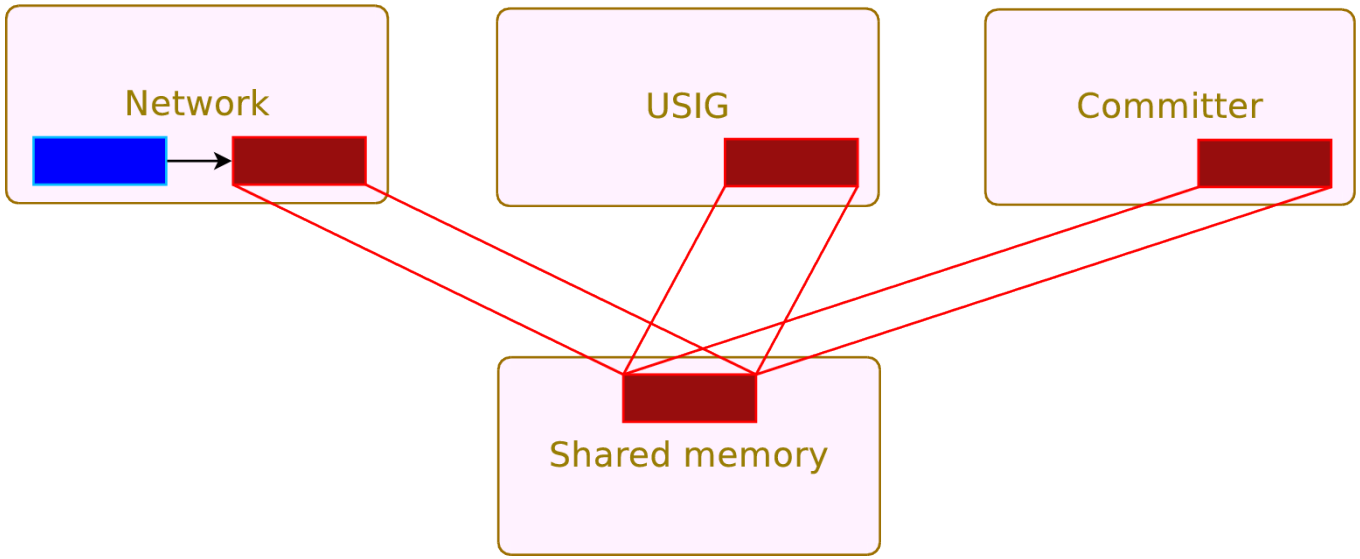
Verification



What guarantees do we get between VST & CompCert?

Implementation

E.g., shared message buffer



Progress so far

securityandtrust.lu

CritiX

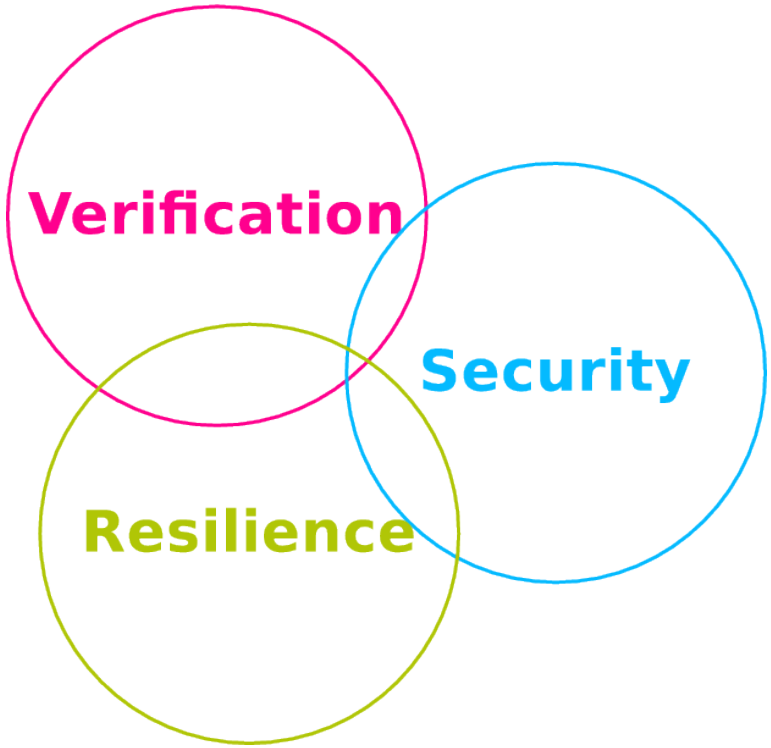
Designing BFT tasks at the Fiasco level

Building and verifying a USIG C task

uni.lu UNIVERSITÉ DU LUXEMBOURG

Thank You!

securityandtrust.lu



We're hiring