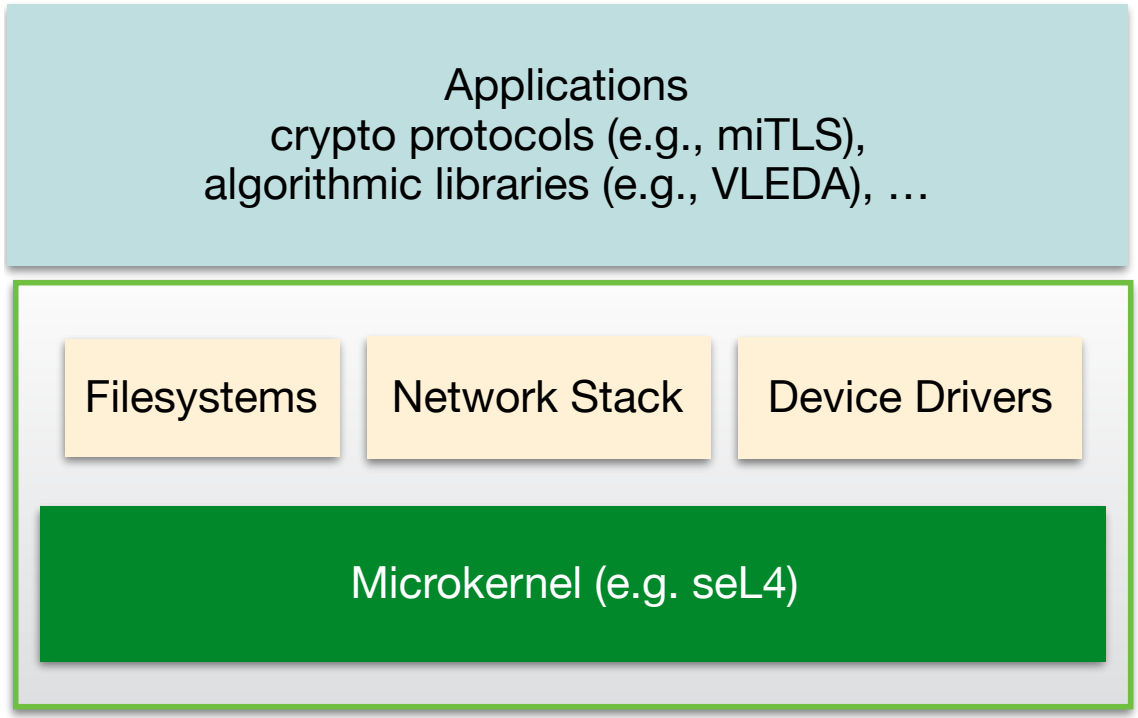
Reducing the Cost of Efficient Secure Verified Software

Christine Rizkallah

Deep Spec Summer School Talk
25 July 2017

Software Verification



- software verification is essential for building secure systems
- verify both **applications** and underlying **systems** code
- attacks on systems code can undermine security
- e.g., attackers can exploit a FS bug to read or corrupt files.

Where Does Verification Stand?

- most notable successes in interactive theorem proving:
  - **CompCert** optimizing C compiler verified in Coq
  - **seL4** microkernel verified in Isabelle/HOL
  - both care about **efficiency** without compromising on **trust**
  - enormous effort: several PhDs (seL4 $\approx 10k$ loc, ≈ 25 p. yrs)
- scale-up verification to other critical software components
- problem: verifying them by brute-force is extremely costly!

Proposal

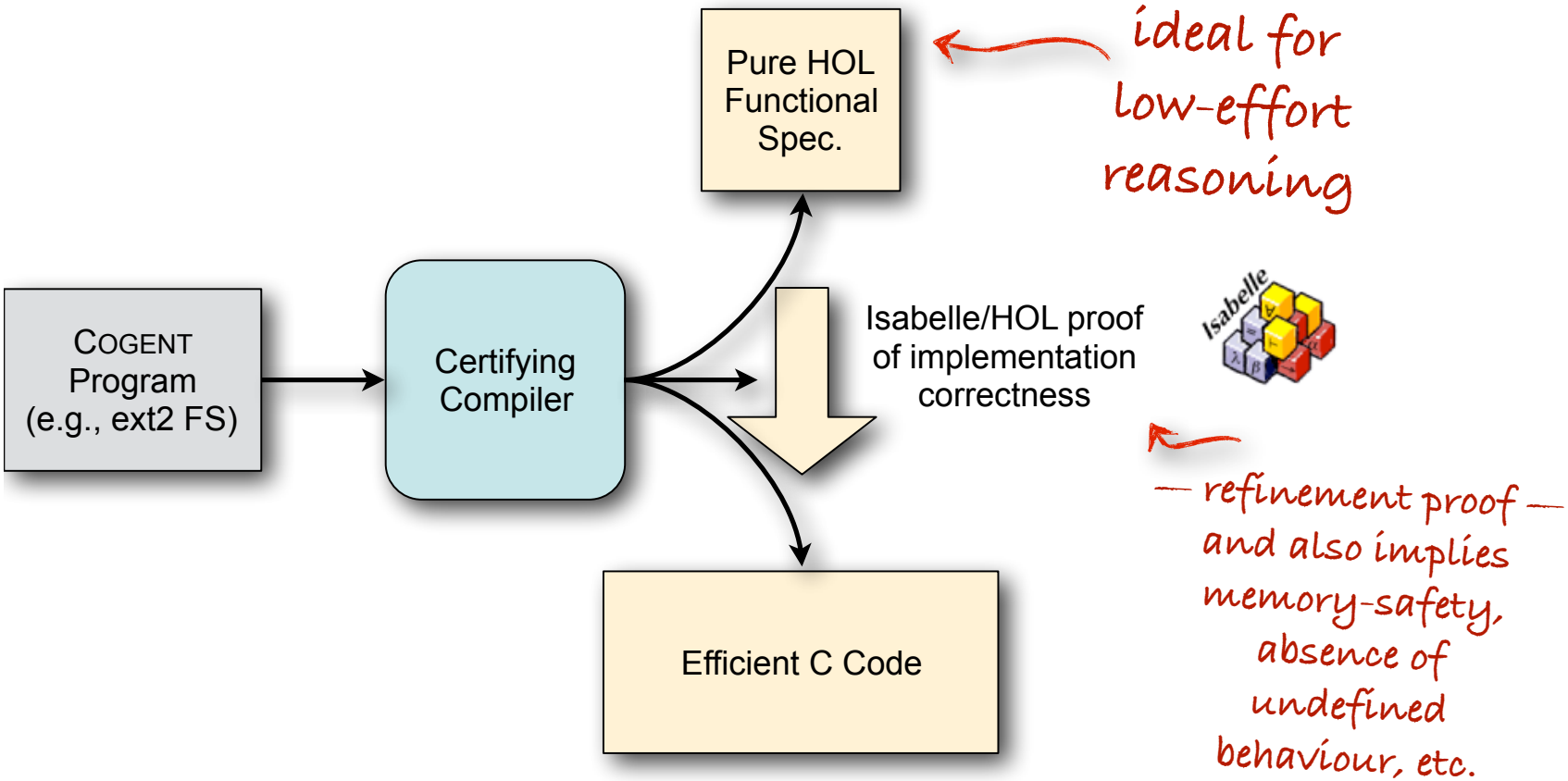
Create methods to **reduce the cost** of verification without compromising on **efficiency** or **trust**.

- to reduce the cost:
- functional languages increase productivity, ease verification
- type systems can auto. enforce safety, security properties
- certifying compilers automate large portion of verification
- **past:** filesystems - **future:** network stack, crypto protocols

Trustworthy Filesystems (Data61)

- open problem; posed **grand challenge** [Freitas et al. 2008]
- many huge filesystems (each $\approx 5k$ loc, linux has ≈ 50)
- elegant method to reduce cost of filesystem verification
- created COGENT: a linearly **typed** restricted **functional language** with **certifying compiler** that co-generates code and proofs

COGENT: Certifying Compiler



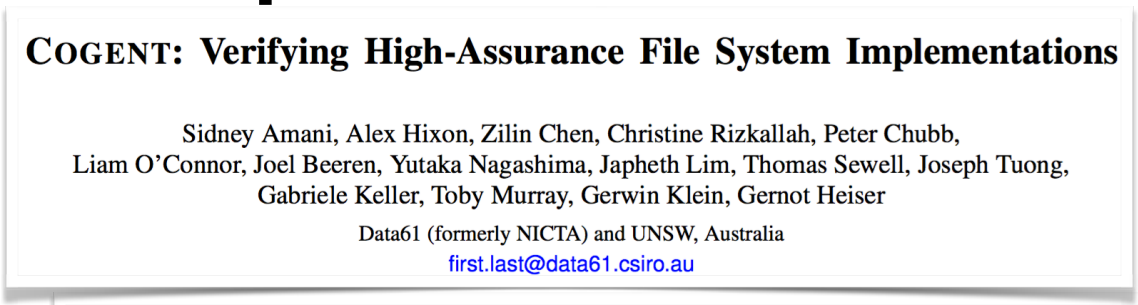
COGENT: Filesystem Verification Results

- built **efficient** COGENT filesystems (ext2 & BilbyFS)
  - linearly typed variables are used exactly once
  - in-place memory updates; no garbage collector
  - each filesystem $\approx 4k$ COGENT loc, plus $\approx 2k$ C loc.
- COGENT **dramatically reduced cost** of verification
  - reasoning about functional spec. instead of directly on C.

    seL4 $\approx$ 1.65 person months per 100 C loc
    filesystems $\approx$ 0.38 person months per 100 COGENT loc
  - thanks to linear types: automatic proof of memory safety

Want to Know More?

Systems [ASPLOS' 16]:



COGENT: Verifying High-Assurance File System Implementations

Sidney Amani, Alex Hixon, Zilin Chen, Christine Rizkallah, Peter Chubb,
Liam O'Connor, Joel Beeren, Yutaka Nagashima, Japheth Lim, Thomas Sewell, Joseph Tuong,
Gabriele Keller, Toby Murray, Gerwin Klein, Gernot Heiser

Data61 (formerly NICTA) and UNSW, Australia
first.last@data61.csiro.au

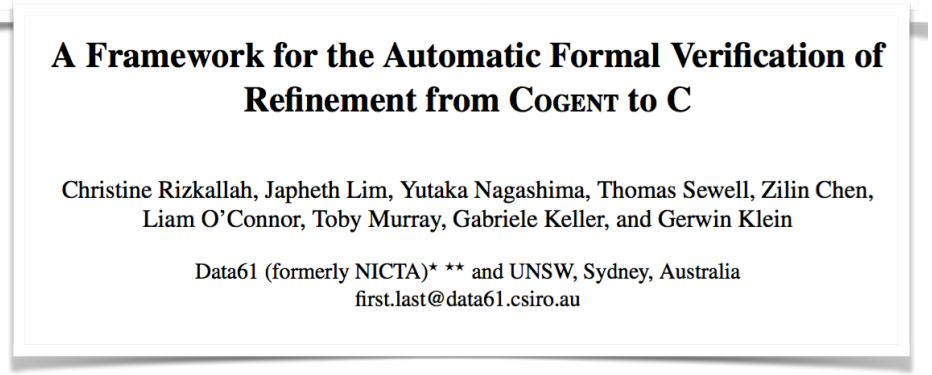
PL [ICFP' 16]:

Refinement Through Restraint: Bringing Down the Cost of Verification

Liam O'Connor[21], Zilin Chen[12], Christine Rizkallah[4], Sidney Amani[12], Japheth Lim[1], Toby Murray[3],
Yutaka Nagashima[1], Thomas Sewell[12], Gerwin Klein[12]

[1] Data61*, Australia [2] UNSW, Australia [3] University of Melbourne, Australia [4] University of Pennsylvania, USA
liamoc@cse.unsw.edu.au firstname.lastname@data61.csiro.au toby.murray@unimelb.edu.au

Verification [ITP' 16]:



A Framework for the Automatic Formal Verification of Refinement from COGENT to C

Christine Rizkallah, Japheth Lim, Yutaka Nagashima, Thomas Sewell, Zilin Chen,
Liam O'Connor, Toby Murray, Gabriele Keller, and Gerwin Klein

Data61 (formerly NICTA)* ** and UNSW, Sydney, Australia
first.last@data61.csiro.au