Towards a Fully Verified File System

Sidney Amani, Leonid Ryzhyk, Toby Murray

Proving the functional correctness of a realistic file system implementation

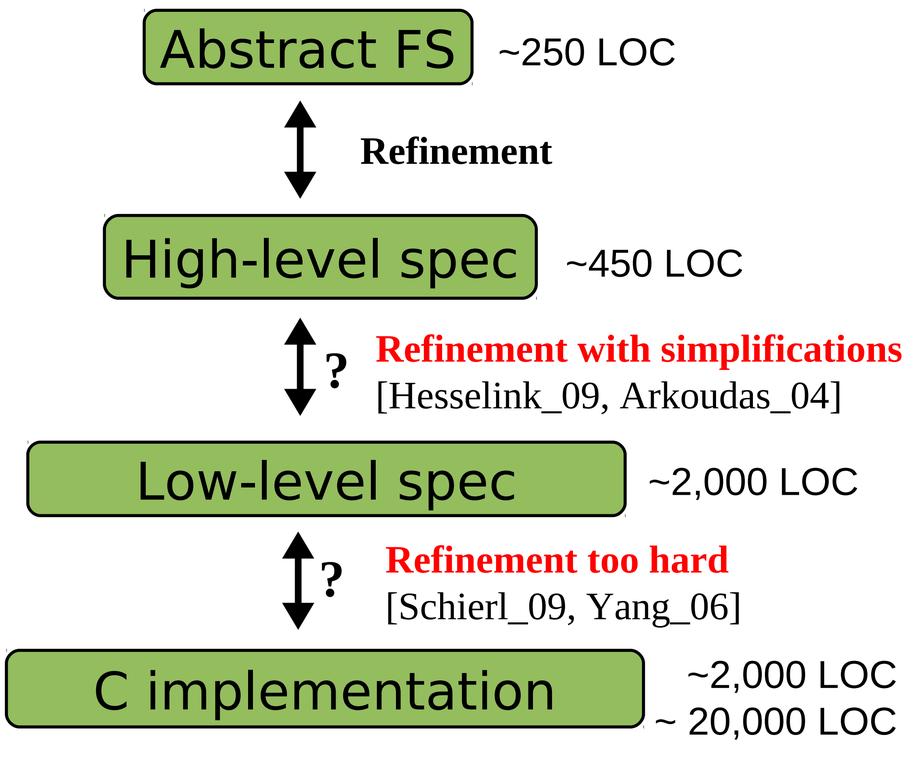
Motivation

- File system defects can lead to disastrous data loss
- Current development techniques do not ensure the absence of implementation flaws

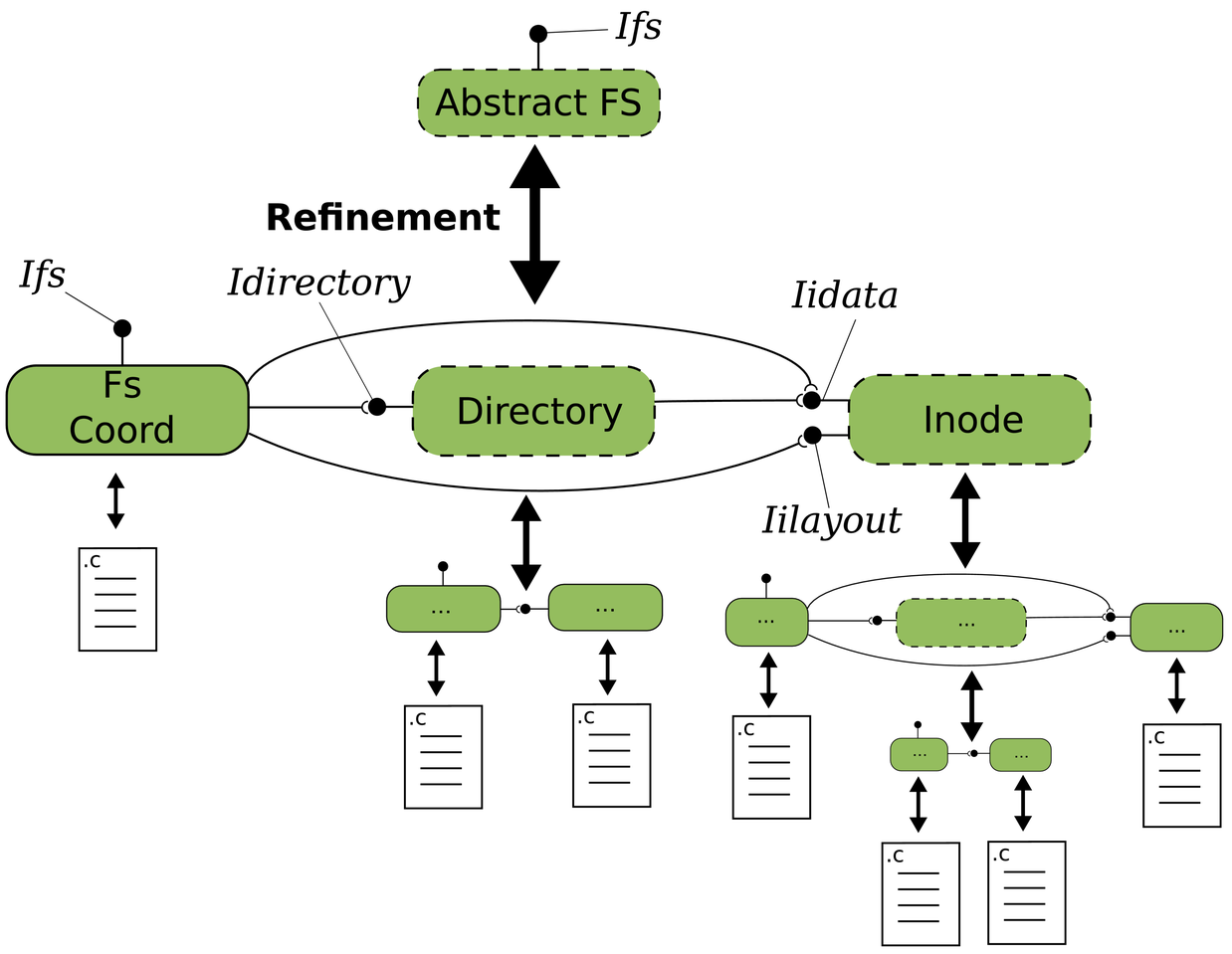
Goal: **Formally verify the functional correctness of a file system implementation**

Problem

- Verifying a file system involves proving that its C implementation refines the abstract specification of file system behaviour
- Refinement proofs are hard for large code bases
- Previous attempts at file system verification could not overcome the complexity of low-level specifications



State-of-the-art in FS verification



File system decomposition example

***Key idea*:**

- Overcoming verification complexity by decomposition
- Introduce implementation details only when refining individual components

Approach:

- Split a specification into multiple components
- Specify well-defined interfaces between them
- Specify the behaviour of each component in the decomposition
- Refine each component individually, by possibly repeating the decomposition process for each of them

Expected research contributions

- First functional correctness proof of a realistic file system implementation
- An approach to file system verification by decomposition