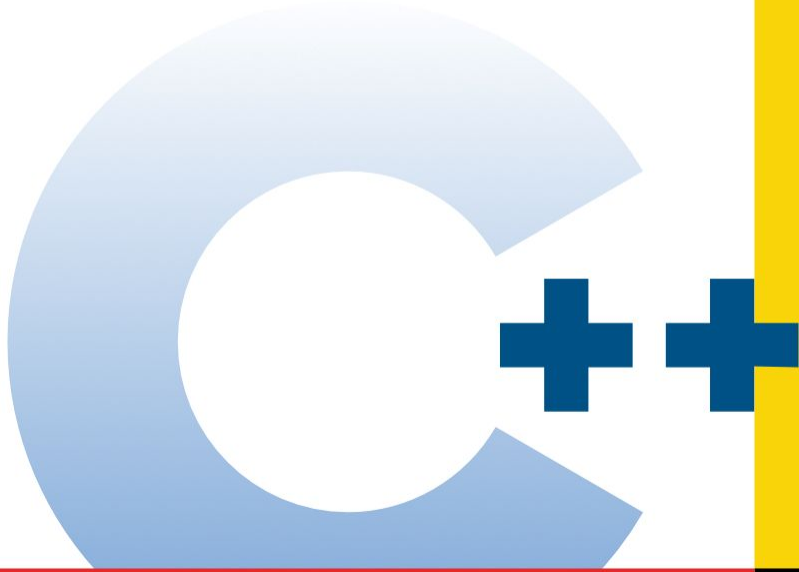
BeC++ SYMPOSIUM

Chris Croft-White

Agentic Time-Travel Debugging

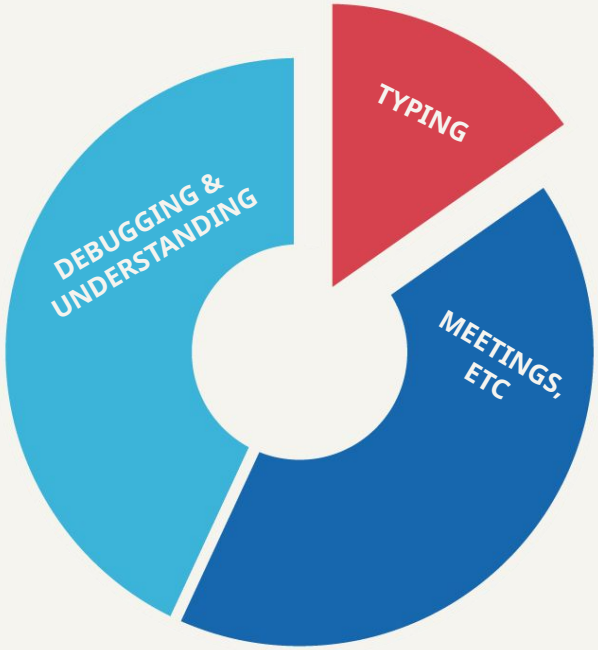


undo

Just

15%

of time spent typing in the code

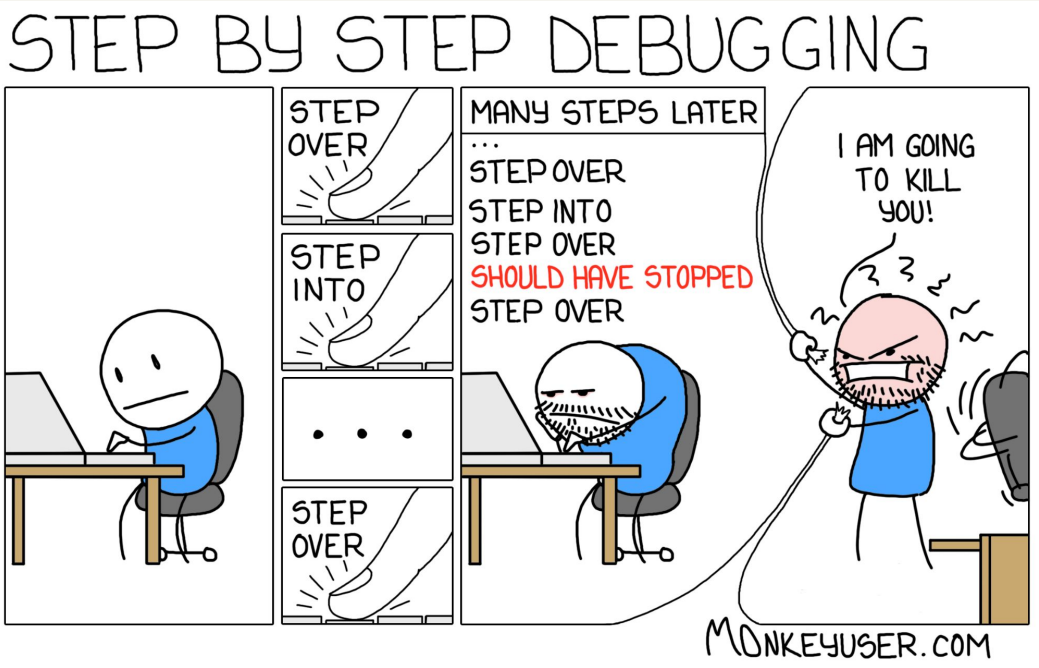


undo

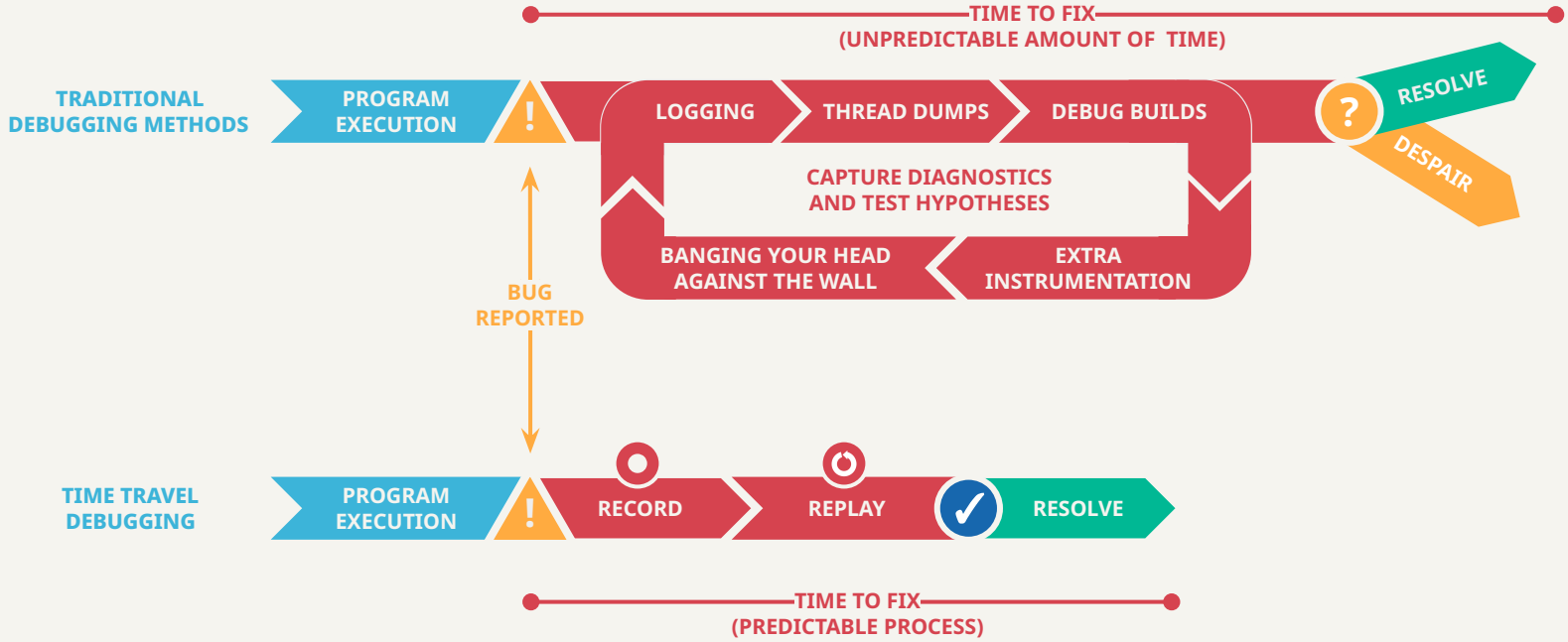
Fantasy

vs

Reality



Why is debugging hard?



undo

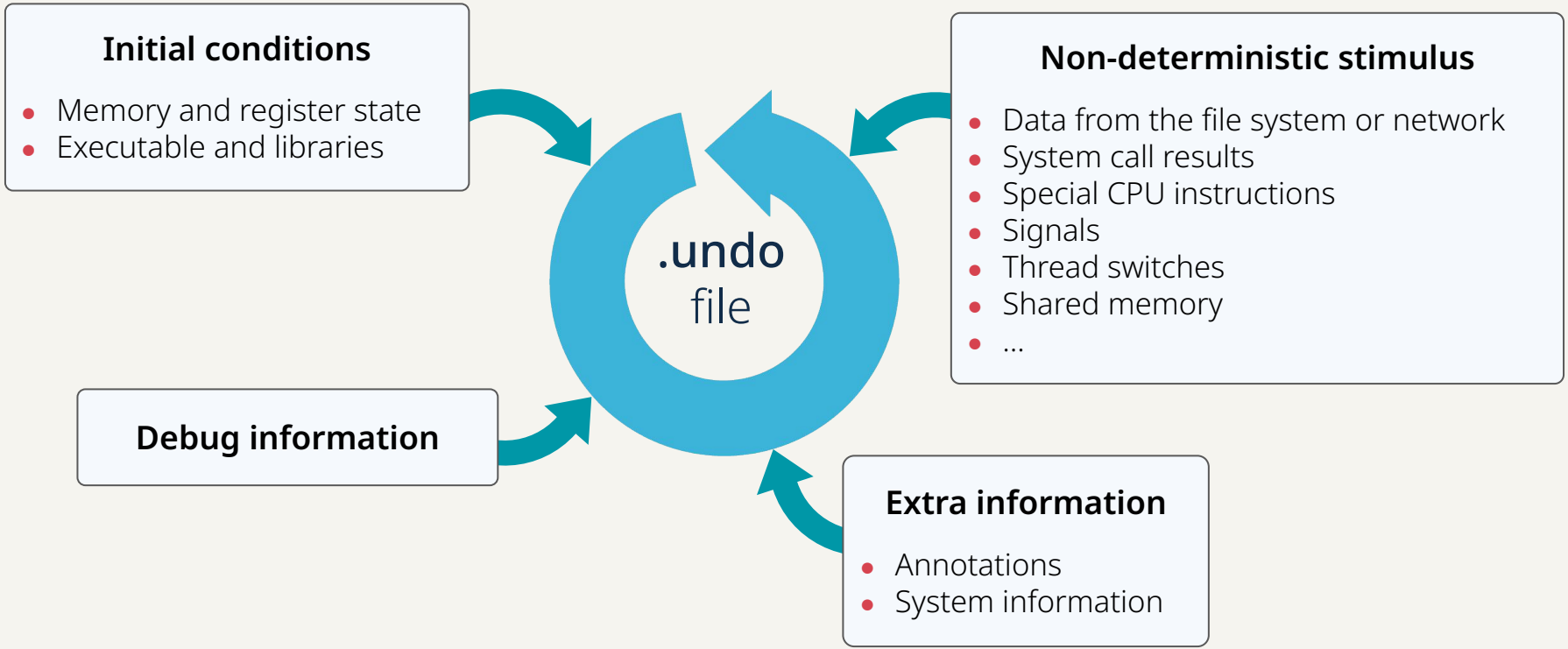
Time-Travel Debugging

undo

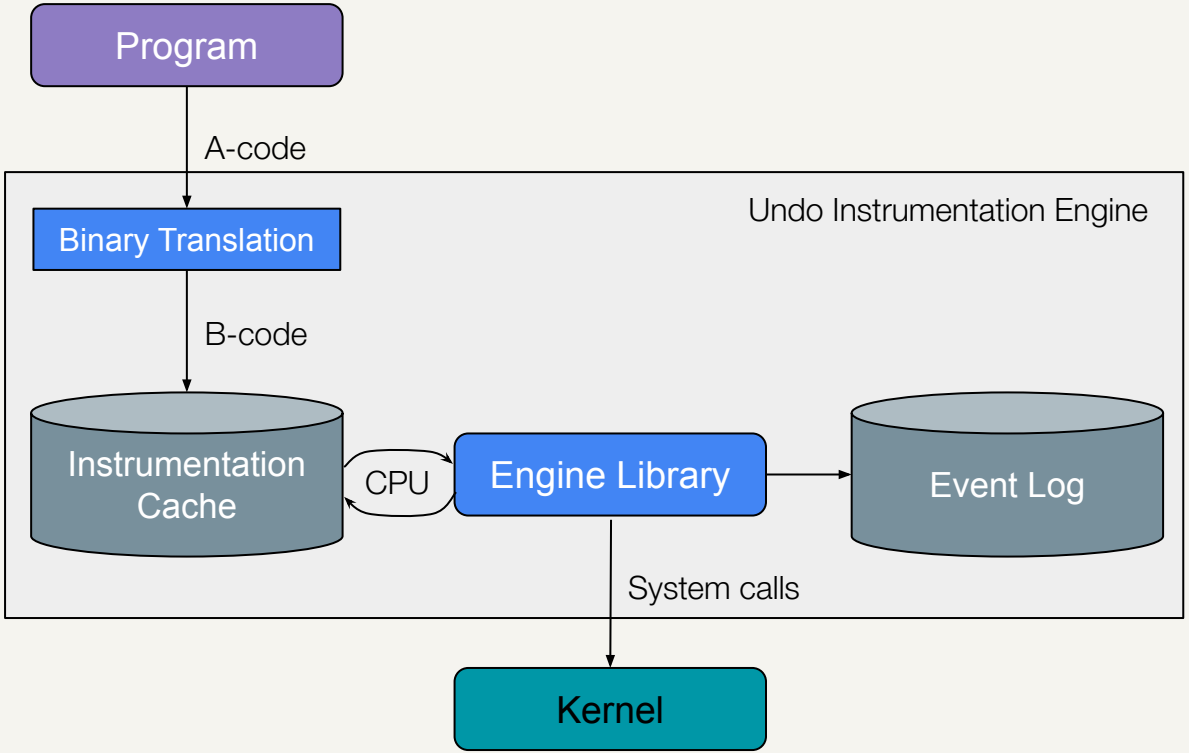
Time-Travel Debugging for C/C++

- **GDB** — built-in reverse execution (slow, but it's there)
- **rr** — open-source record & replay for Linux
- **WinDbg TTD** — Microsoft's solution for Windows
- **UDB/LiveRecorder** — Undo's commercial offering for Linux

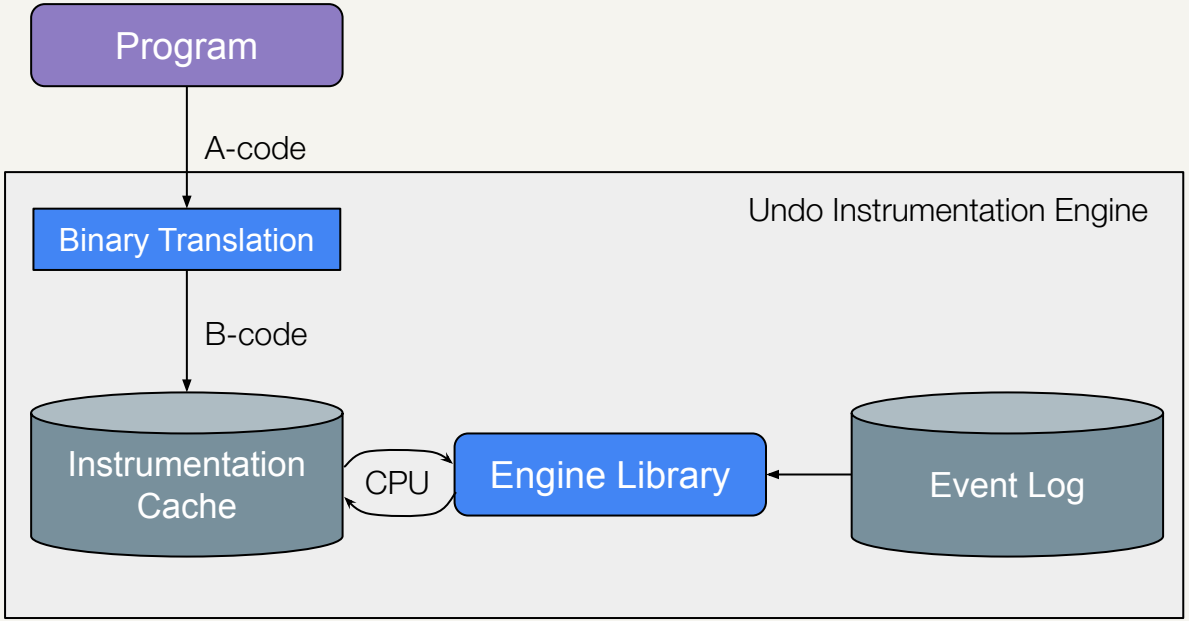
What's saved in an Undo recording?



How Undo records



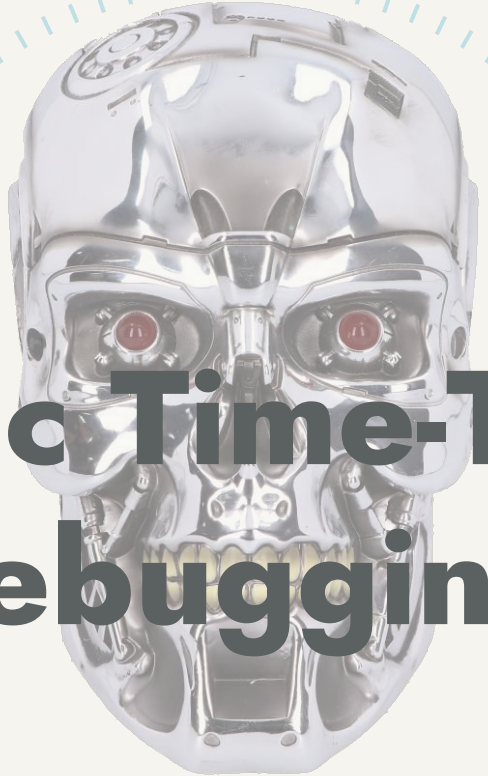
How Undo replays



undo

Reanimating zombies

undo



Agentic Time-Travel Debugging

Disclaimer: Undo bears no responsibility for Skynet confusing humans for software bugs.

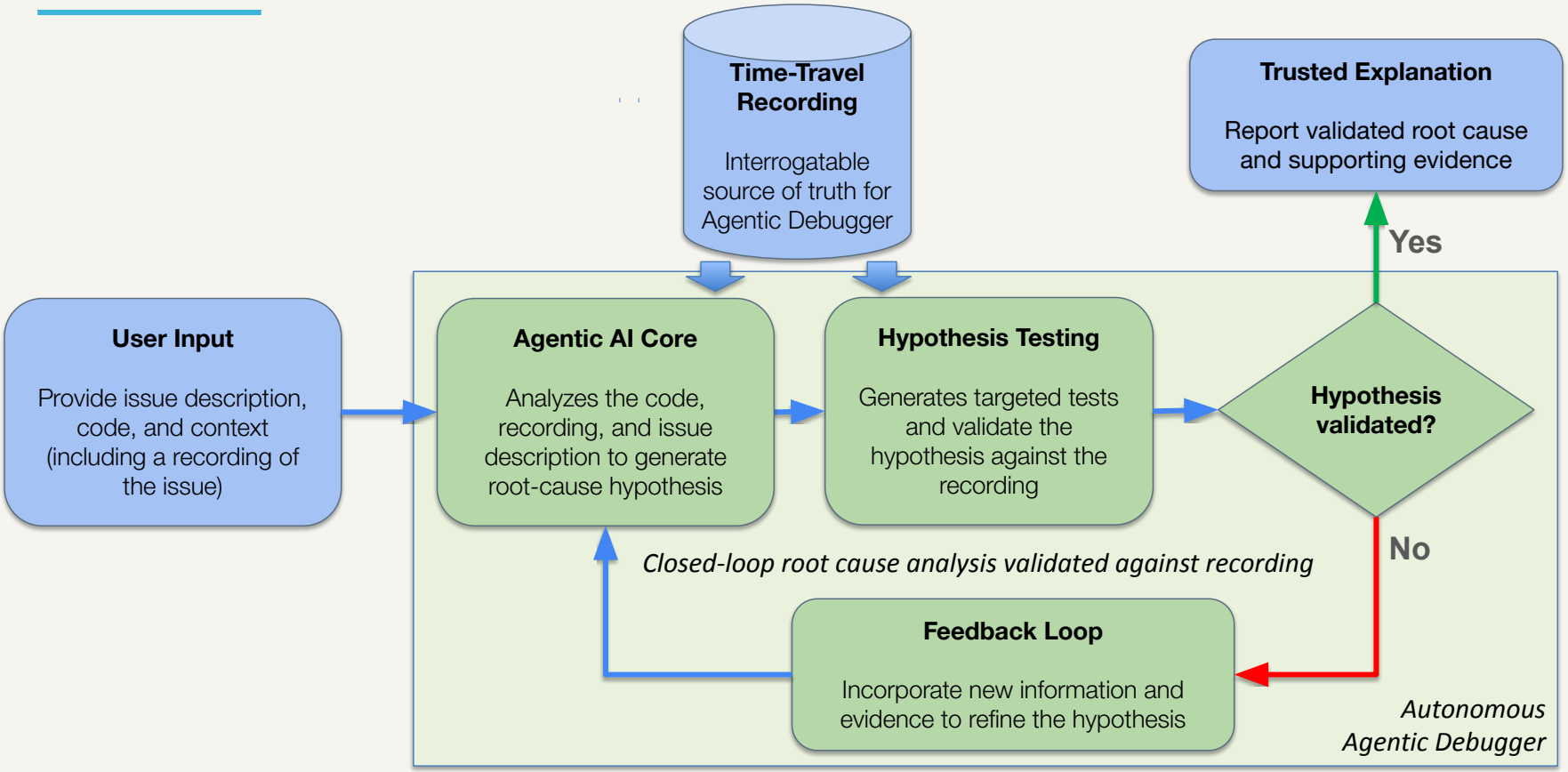
undo

AGENTIC DEBUGGING

- Give AI access to time travel debugging tools and let it **autonomously** root cause issues for you
- The recording provides:
  - Means to explore the code in a directed way
  - Rich information about dynamic behaviour
  - A *ground truth* for both the AI and the developer

Closing the loop to enable reliable agentic root cause analysis

Agentic Time-Travel Debugging



undo

Playing with trains (agentically)

undo

Chris Croft-White

Staff Solutions Architect

chris@undo.io