

Unleashing Open-Source Silicon

Steve Hoover
Redwood EDA

Agenda

- Open-source silicon
 - challenges
 - status
- Cloud FPGAs
 - why they are so important
 - what they lack
 - our solution -- 1st CLaaS
 - *[demo video]*

Open-Source Silicon - What's holding us back?

1. Implementation detail
2. Access to tools
3. Access to h/w

...and the Barriers are Breaking Down!!!

1. Implementation detail

- HLS
- Hardware construction

2. Access to tools

- Complete open-source FPGA design flows now exist
- US Government funding open-source EDA for ASICs and boards

3. Access to h/w

- Cloud FPGAs are an answer!

I'm doing my part

1. Impl. detail
 - TL-Verilog
2. Access to tools
 - makerchip.com
3. Access to h/w
 - Integration w/
Cloud FPGAs

The screenshot displays the makerchip IDE interface. The top navigation bar includes 'PROJECT', 'TUTORIALS', and 'HELP'. The main workspace is divided into three panels:

- EDITOR:** Shows Verilog code for a pipelined Pythagorean Theorem calculator:

```
@1  
Saa_sq[7:0] = $aa[3:0] ** 2;  
Sbb_sq[7:0] = $bb[3:0] ** 2;  
@2  
$cc_sq[8:0] = $aa_sq + $bb_sq;  
@3  
$cc[4:0] = sqrt($cc_sq);
```
- DIAGRAM:** Shows a logic diagram with three pipeline stages labeled @1, @2, and @3. Stage @1 contains two multipliers for \$aa and \$bb. Stage @2 contains an adder for \$cc_sq. Stage @3 contains a sqrt block for \$cc. Data flows from left to right through the pipeline.
- WAVEFORM:** Shows a timing diagram with a clock signal 'clk' and signals for 'TLV' and '|calc'. The '|calc' signals are \$aa, \$bb, \$aa_sq, \$bb_sq, \$cc_sq, and \$cc. The waveform shows the data values at each clock cycle.

Below the diagram is a caption: "Figure 1: Pipelined Pythagorean Theorem Logic".

Below the caption is a text description: "This pipeline is 3 cycles deep. It has a throughput of one transaction per cycle, where a transaction performs one Pythagorean Theorem calculation per cycle."

Impact of Open-Source Silicon and Cloud Silicon

- More dramatic transformation than open-source and cloud software
 - Silicon is BIG industry
 - Open-source software set the stage
- “Golden age of computing” -> demand for many new architectures
- The transformation:
 - ASICs: Chip makers => data center companies
 - FPGAs: Chip makers => startups
 - Open-source FPGA => open-source ASIC

4th Barrier - Patents

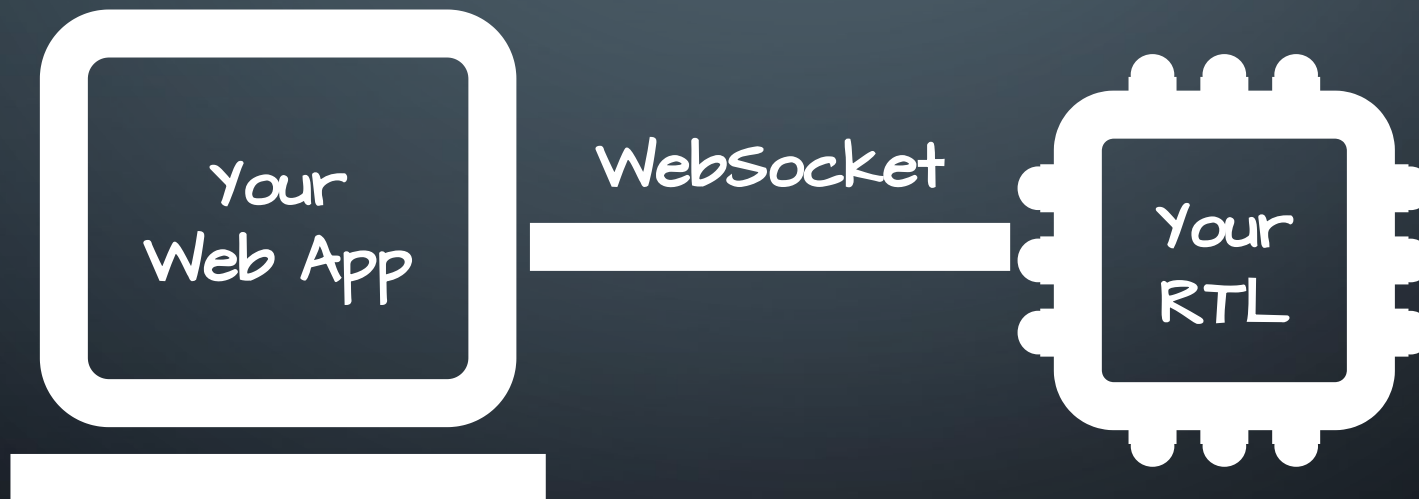


Cloud FPGAs

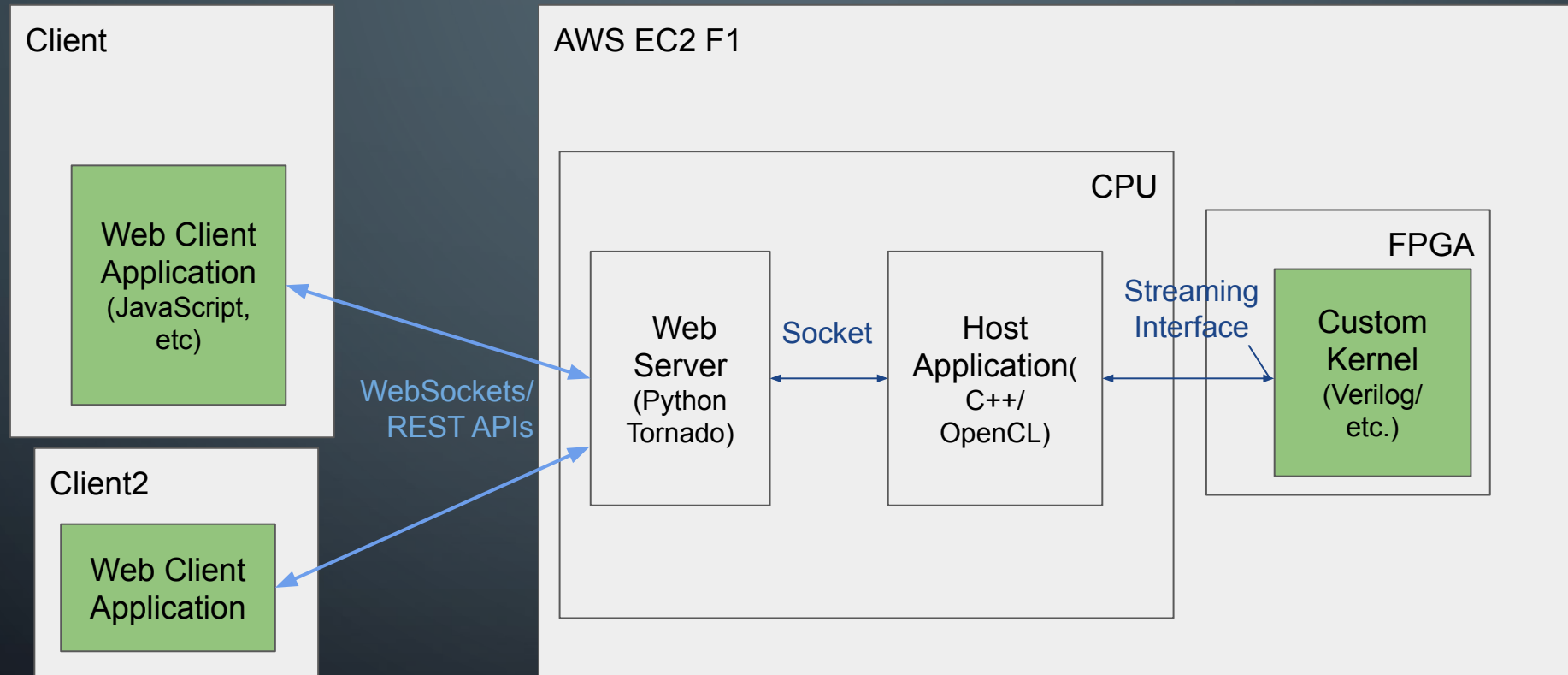
- Hardware platforms exist (e.g. AWS F1)
- Software ecosystem is immature:
 - built for local development
 - built for professionals
 - dropped into the cloud
- Little focus on:
 - developing in cloud
 - small-scale development
 - integrating FPGAs w/ cloud applications

1st CLaaS

- CLaaS: Custom Logic as a Service
- Focused on ease of development
- Communication layer from web app to H/W kernel

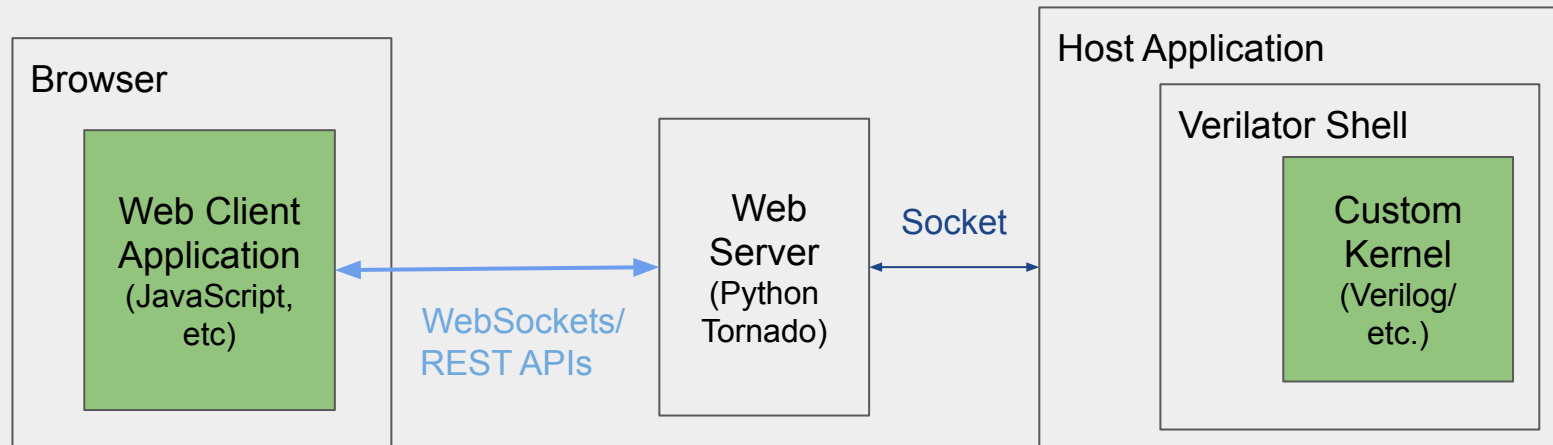


Hardware-Accelerated Web Application



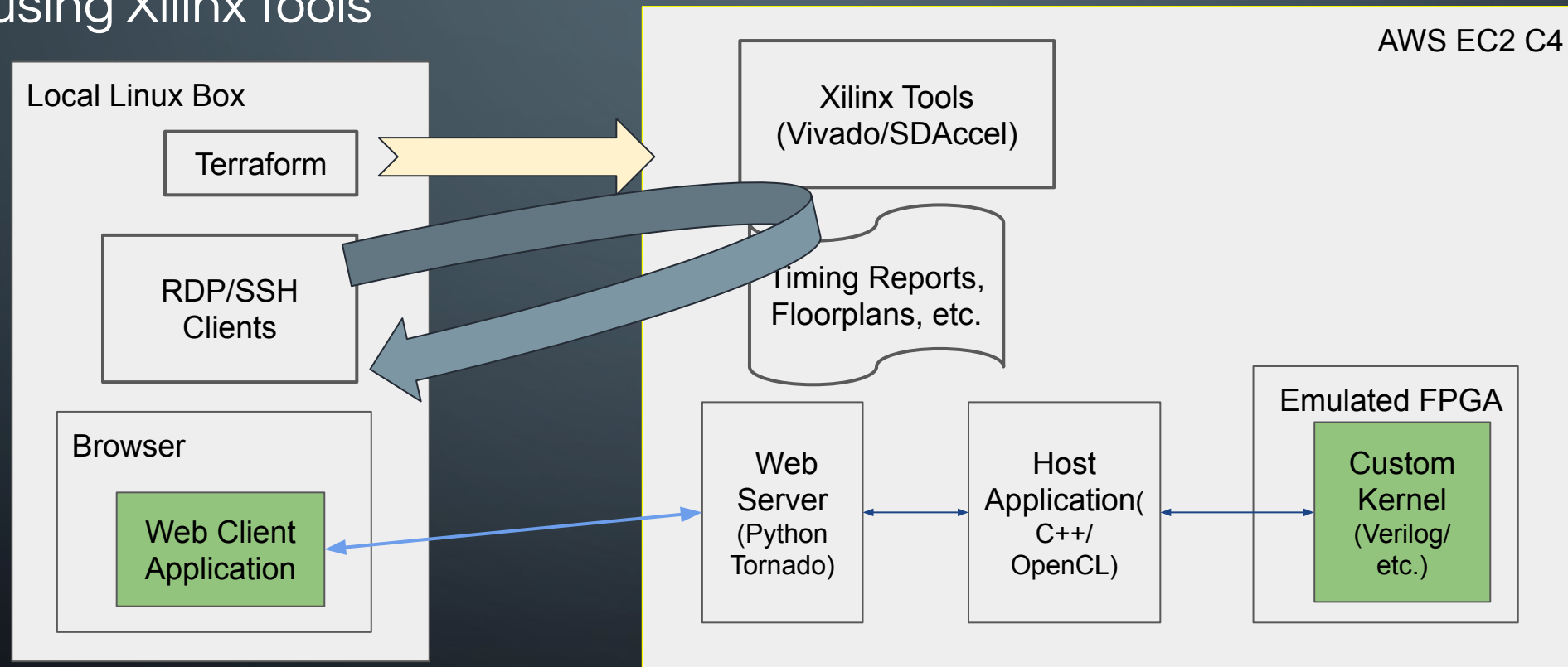
Local Development

Linux box



Implementation Instance

- For optimizing implementation using Xilinx tools

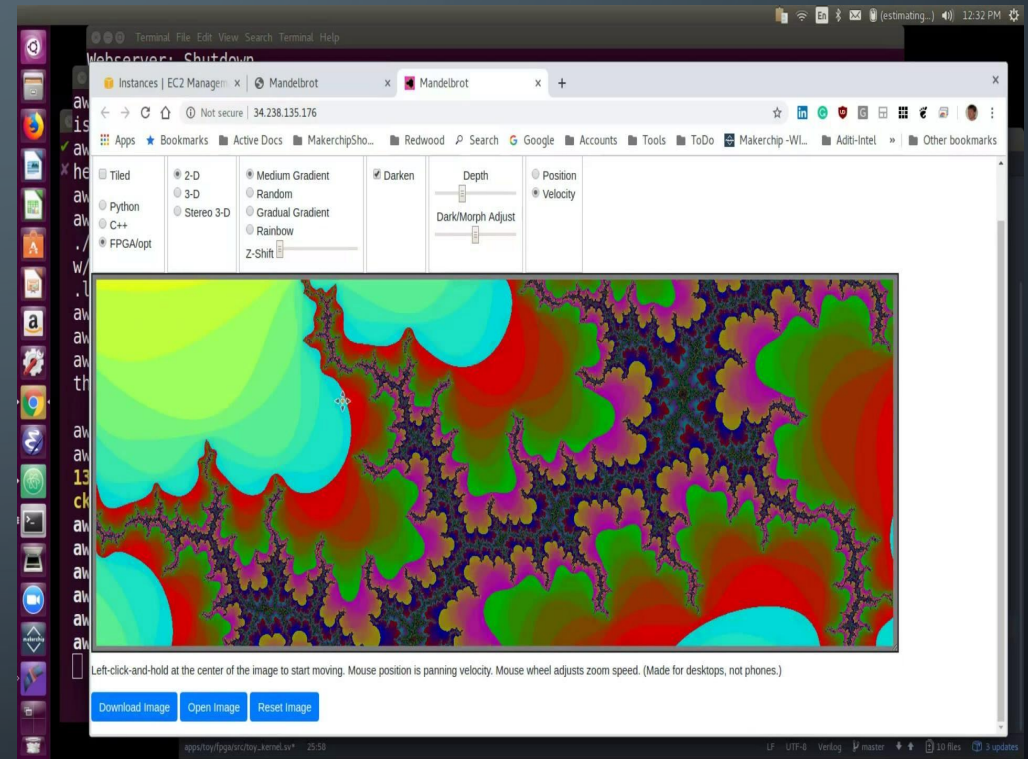


1st CLaaS Usage Examples

3-Part Screen-Capture Video

1. Install 1st-CLaaS locally and run Mandelbrot Demo w/ Verilator
2. Install 1st-CLaaS on F1 and run Mandelbrot Demo w/ FPGA
3. Create simple custom kernel and talk w/ Verilated kernel

Time: 11 min. (edited to 5:30 min.)



Conclusion

1st CLaaS enables open-source FPGA development that is:

- integrated with software development
- low cost
- nearly as easy as open-source software development

No more barriers!

Open-source silicon will hit hard!