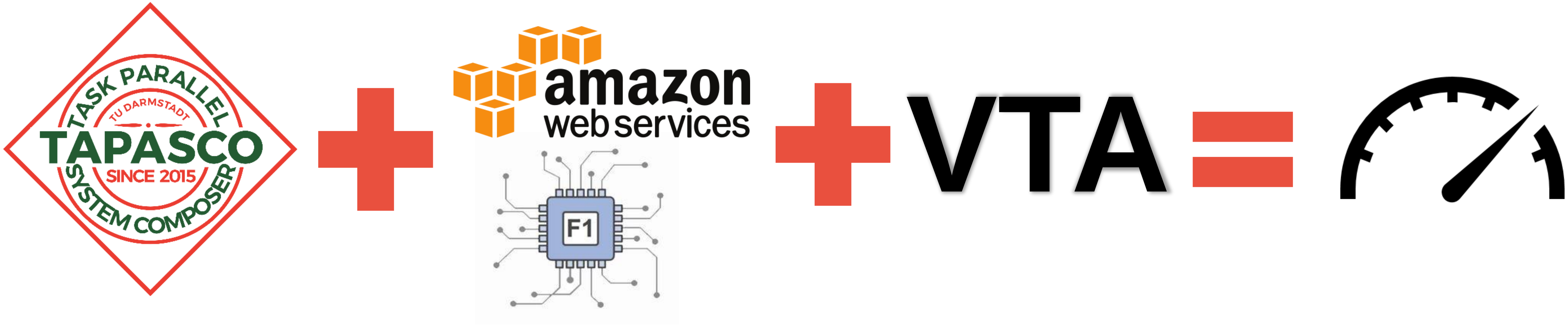
TaPaSCo: Task-Parallel System Composer for FPGAs
Deploy VTA on More Platforms

Samuel Groß, Florian Stock, Carsten Heinz, Jaco A. Hofmann,

Lukas Sommer, Lukas Weber, Andreas Koch

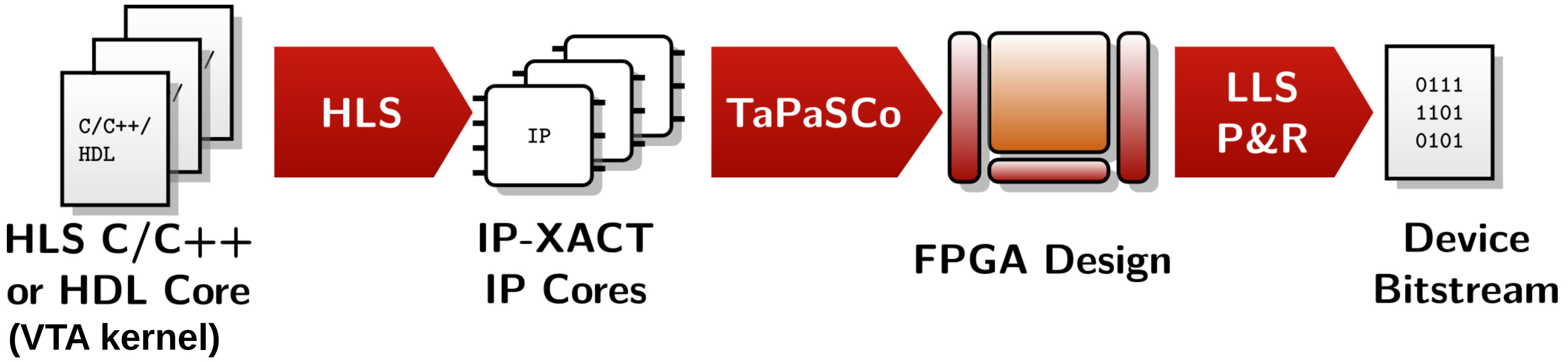
Embedded Systems and Applications Group, TU Darmstadt



TaPaSCo Framework

- Builds complete FPGA SoC-designs from HLS kernels or custom HDL cores
- Automates Design-Space Exploration to determine best system composition
- Supports wide variety of Xilinx platforms
- Includes software API for dispatching compute tasks to FPGA
- Available as free & open-source software

TaPaSCo Design Flow



Core name

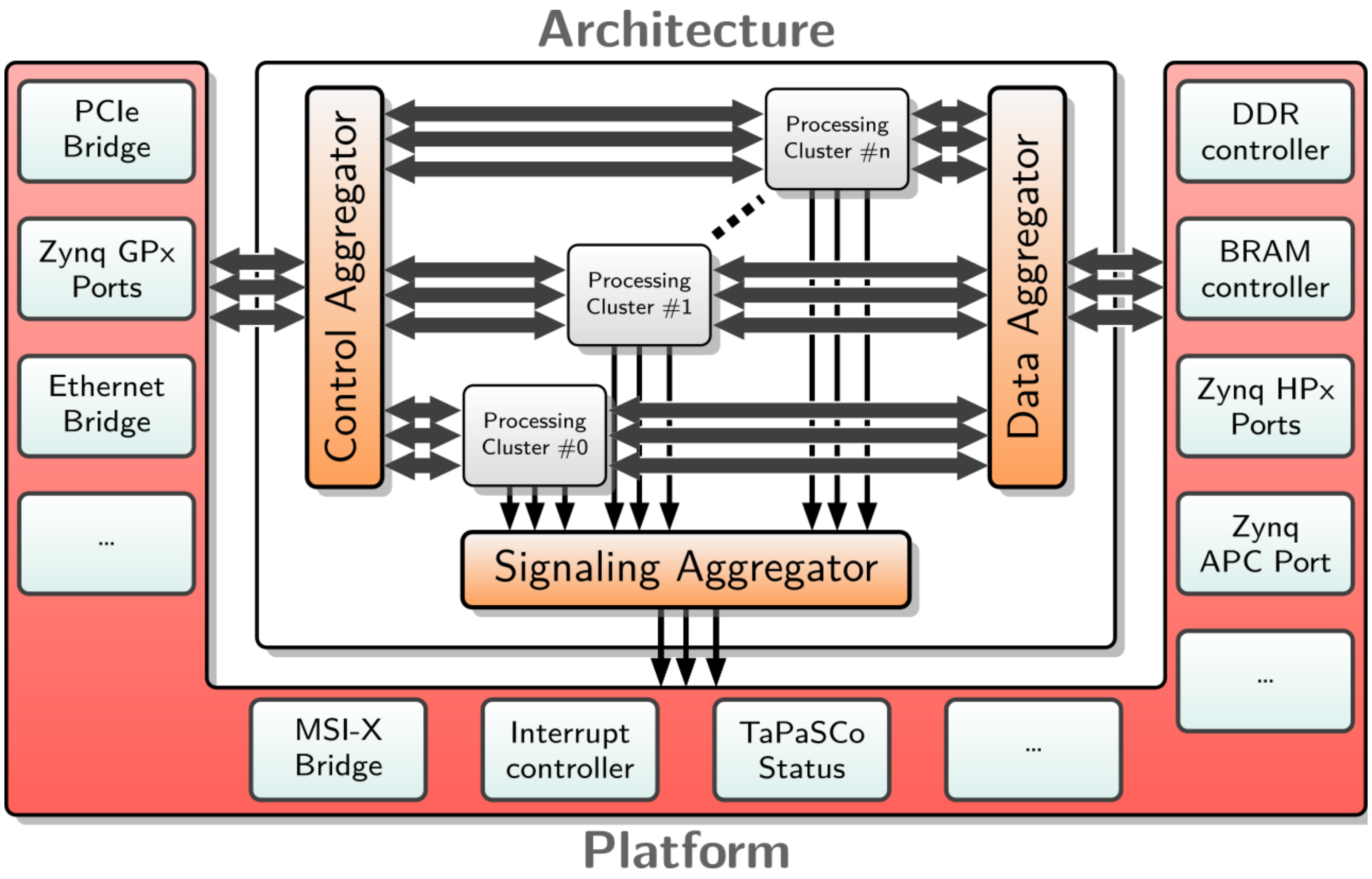
Design frequency

```
tapasco compose [vta x 2, sobel x 3] @ 100 MHz -p vc709
```

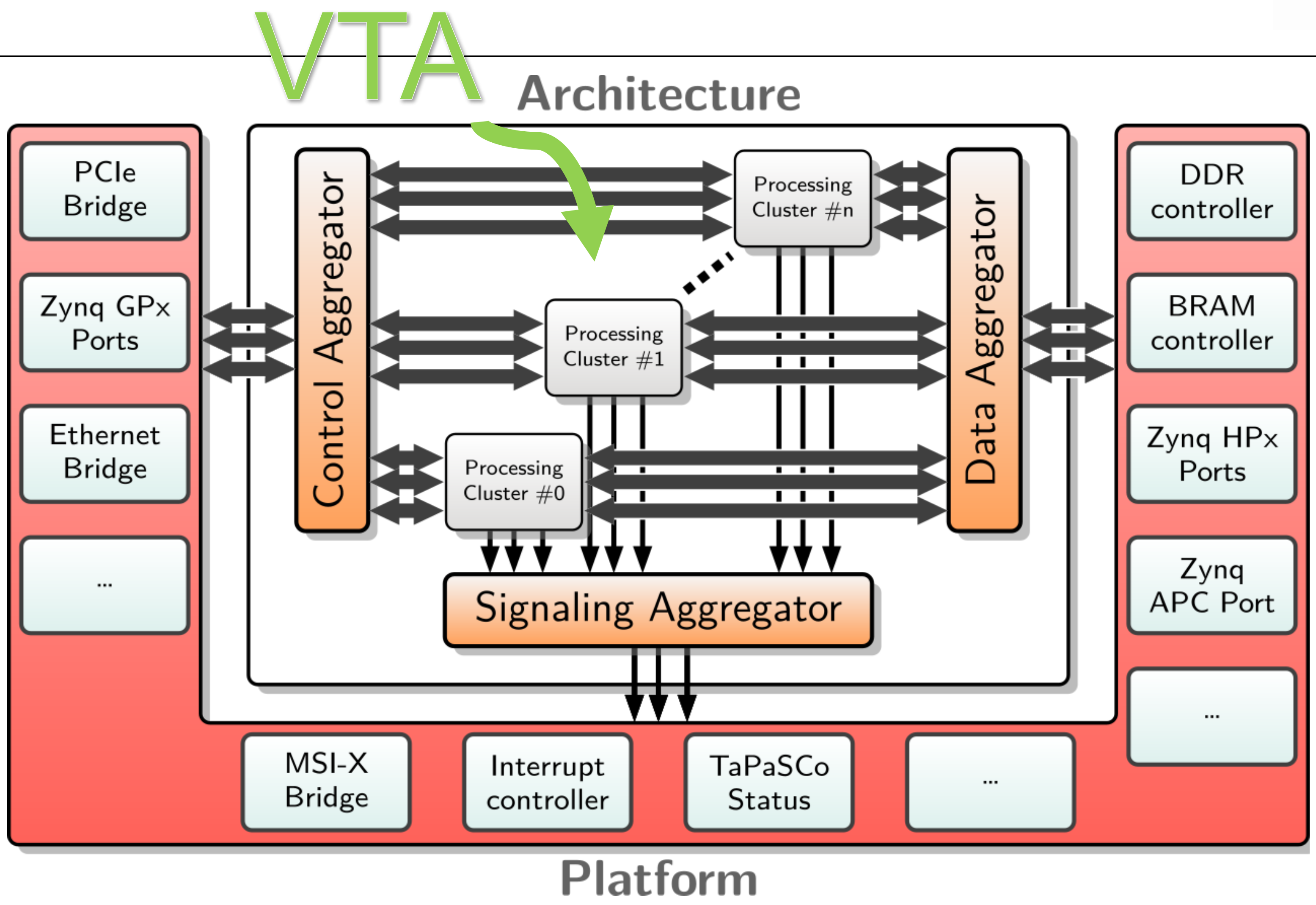
Core count

Platform

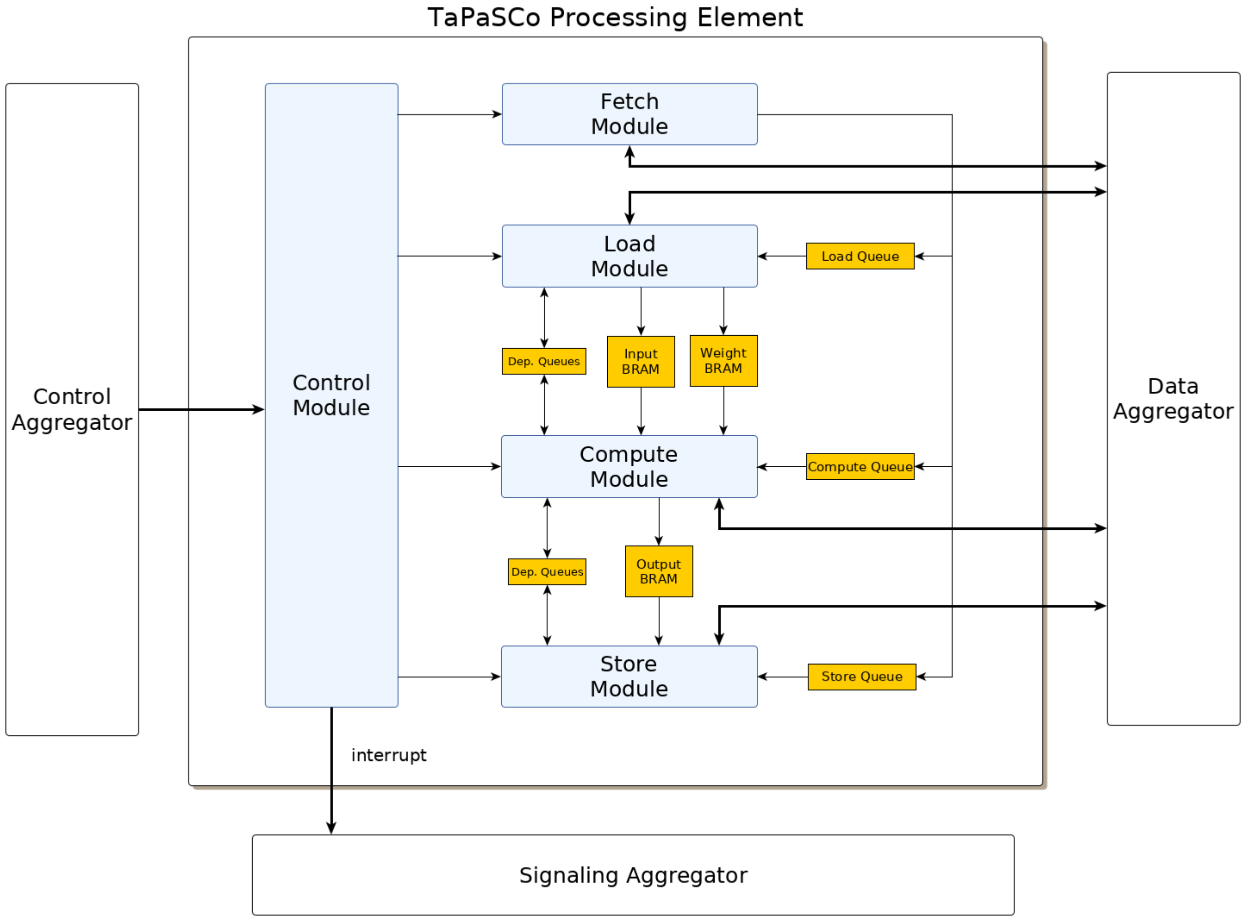
TaPaSCo Architecture



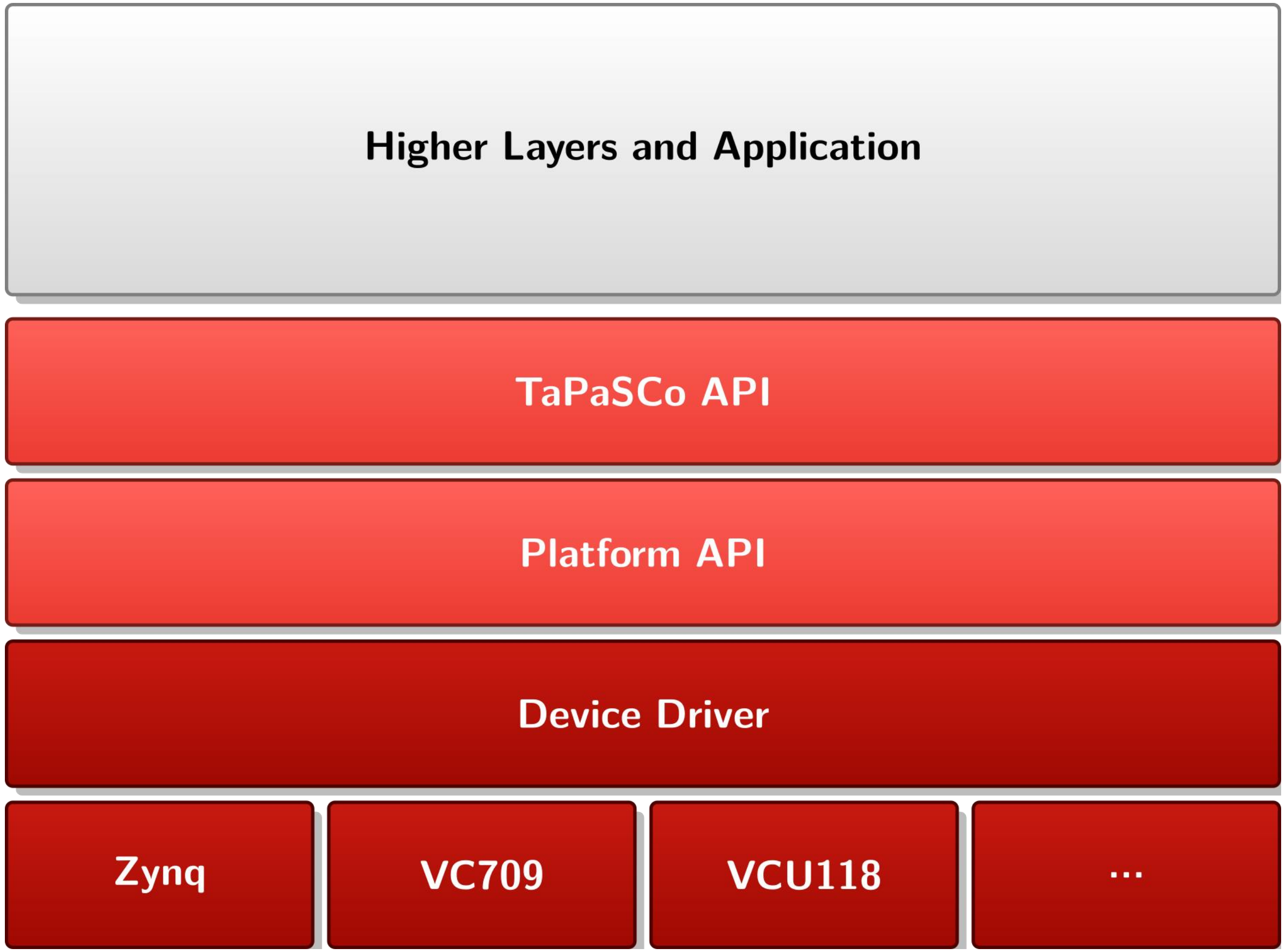
TaPaSCo Architecture



TaPaSCo – VTA PE



TaPaSCo Software API



Higher Layers and Application

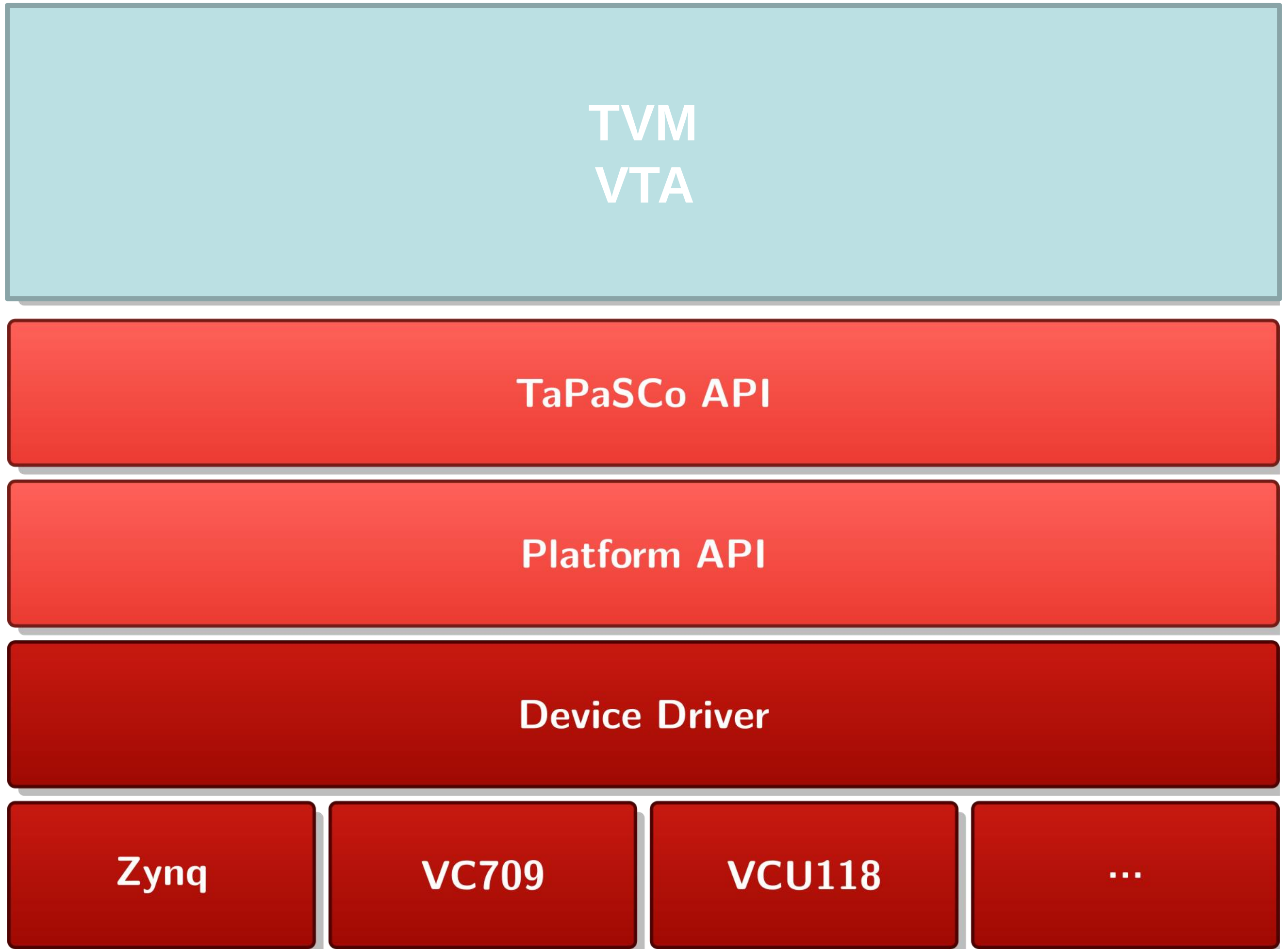
TaPaSCo API

Platform API

Device Driver

Zynq | VC709 | VCU118 | ...

TaPaSCo Software API



TaPaSCo Platforms

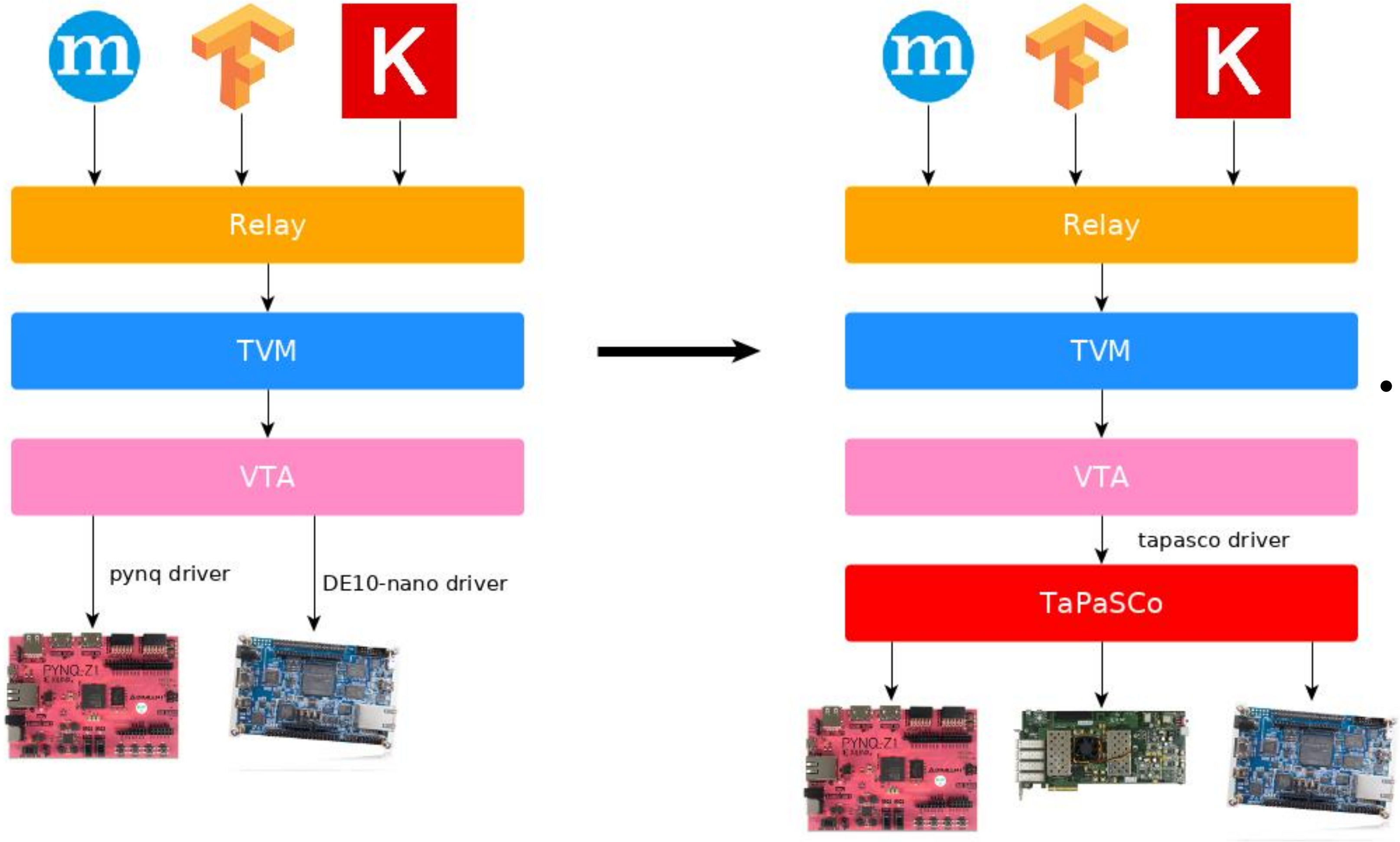
Datacenter

- Xilinx Alveo U250
- Xilinx Virtex UltraScale+ VCU1525
- Xilinx Virtex UltraScale+ VCU118
- Xilinx Virtex UltraScale VCU108
- Digilent NetFPGA SUME
- Xilinx Virtex VC709
- Amazon AWS F1 instance

Edge Devices

- Xilinx Zynq UltraScale+ MPSoC ZCU102
- Xilinx Zynq SoC ZC706
- AVNET ZedBoard
- Digilent Pynq-Z1

TVM/VTA Stack



Relay

TVM

VTA

pynq driver

DE10-nano driver

Relay

TVM

VTA

tapasco driver

TaPaSCo

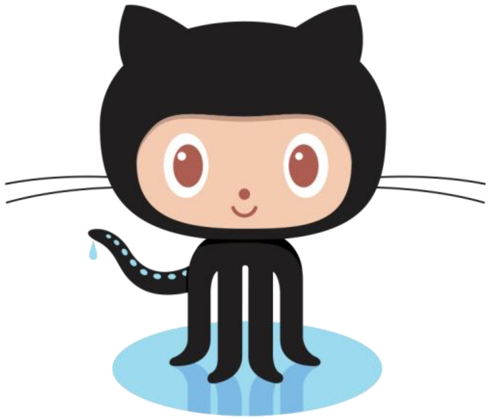
- Advantages:
  - One generic driver
  - Many different platforms
  - Multiple VTA instances (WIP)
  - Larger VTA instances (WIP)

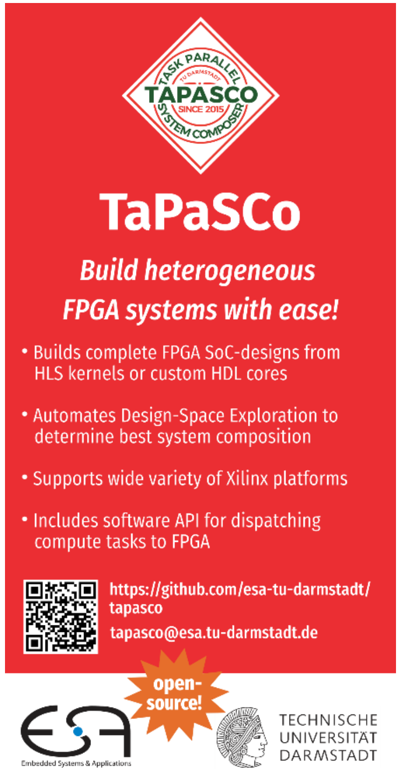
Shameless Advertising

Start to build your own AWS F1 accelerator system using TaPaSCo!

Download TaPaSCo from Github:

`github.com/esa-tu-darmstadt/tapasco`





TaPaSCo

Build heterogeneous FPGA systems with ease!

- Builds complete FPGA SoC-designs from HLS kernels or custom HDL cores
- Automates Design-Space Exploration to determine best system composition
- Supports wide variety of Xilinx platforms
- Includes software API for dispatching compute tasks to FPGA

https://github.com/esa-tu-darmstadt/tapasco

tapasco@esa.tu-darmstadt.de

open-source!
Embedded Systems & Applications
TECHNISCHE UNIVERSITÄT DARMSTADT

ADDITIONAL BONUS SLIDES

TaPaSCo Software API – Example

Wrap information about data-transfer

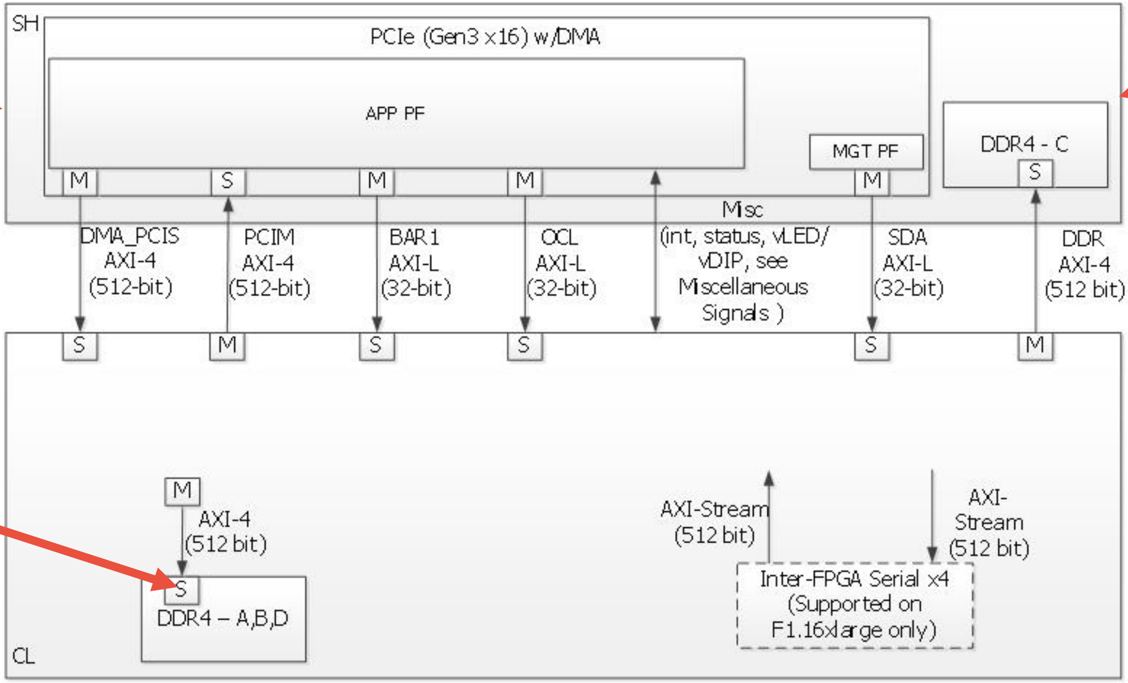
```
Tapasco tapasco;
auto a_wrapped = makeWrappedPointer(a.data(), a.size());
auto b_wrapped = makeWrappedPointer(b.data(), b.size());


auto job = tapasco.launch(SIMPLE_HLS_ID,
  makeInOnly(a_wrapped), makeOutOnly(b_wrapped));
job();
```

Launch FPGA execution

Provide information about data-transfer direction

TaPaSCo in the Cloud

- Amazon deploys Xilinx VU9+ FPGAs in AWS EC2 F1 instances
- Most of the FPGA logic freely programmable, all interfaces routed through fixed *Shell* provided by Amazon



Shell

DDR4 channel

3 Optional DDR4 channels

Custom logic

Image source: Amazon

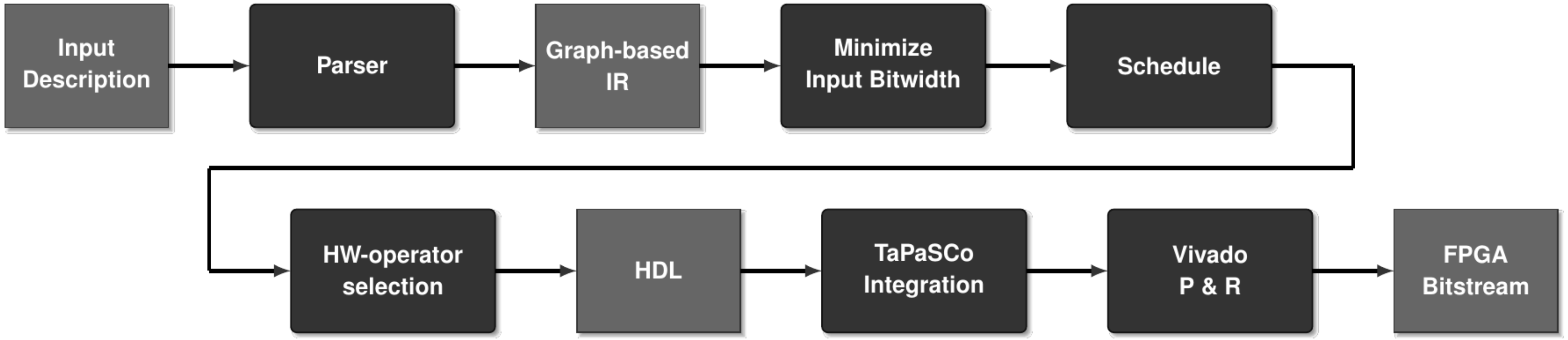
TaPaSCo in the Cloud - Challenges

- *Shell* provides only a few frequencies, TaPaSCo supports arbitrary design frequencies
  - Include custom clock controller in programmable logic
- DMA engine in *Shell* provides only limited throughput
  - Replace with TaPaSCo‘s own DMA engine
- *Shell* provides only 16 interrupts, not enough for TaPaSCo architecture
  - Include custom interrupt controller for translation
- Memory controllers for 3 DDR channels have to be placed in custom logic
  - Carefull timing necessary

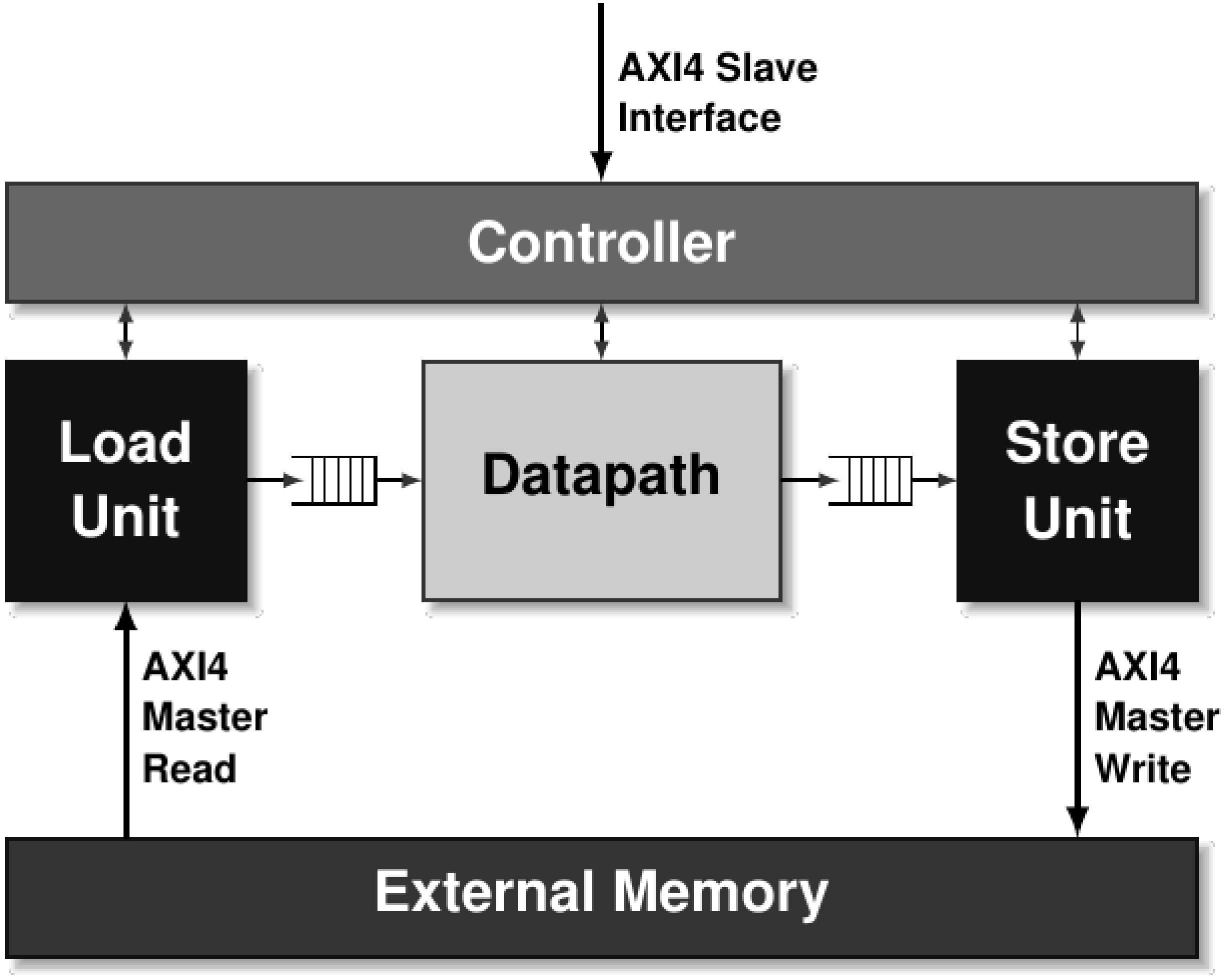
TaPaSCo in the Clouds – Conclusion

- Completely automated toolflow to generate SoC-design from HLS code or custom HDL core for Amazon AWS EC2 F1 FPGA instances
- Generates ready-to-use Amazon FPGA Image (AFI)
- Supports up to four independent memory channels
- Easy-to-use software API for interfacing with FPGA accelerator
- Open-source available!

Existing FPGA Acceleration Toolflow



Existing FPGA Accelerator Core


AXI4 Slave
Interface
Controller
Load
Unit
Datapath
Store
Unit
AXI4
Master
Read
AXI4
Master
Write
External Memory