F. Abel – IBM Research Europe

cloudFPGA

Promoting FPGAs to become 1st class-citizens in datacenters

Workshop: "*The Future of FPGA-Acceleration in Cloud and Datacenters*", FCCM 2020, May 6

Prologue – What are we trying to solve?

Goal ➔ Deploy FPGAs at <u>large scale</u> in hyperscale DCs

⮱ 1-10s of thousands per DC

Requirements

- Server commodity & homogeneity
- Decrease in cost and power
- Easy to manage and to deploy
- On-demand acceleration
- High utilization + workload migration
- Security, virtualization, orchestration
- Hybrid → public & private
- Flexible → IaaS, PaaS, FaaS
- Clusters → #accelerators per server
- Community → #APPs, # developers

Prologue – What are we trying to solve?

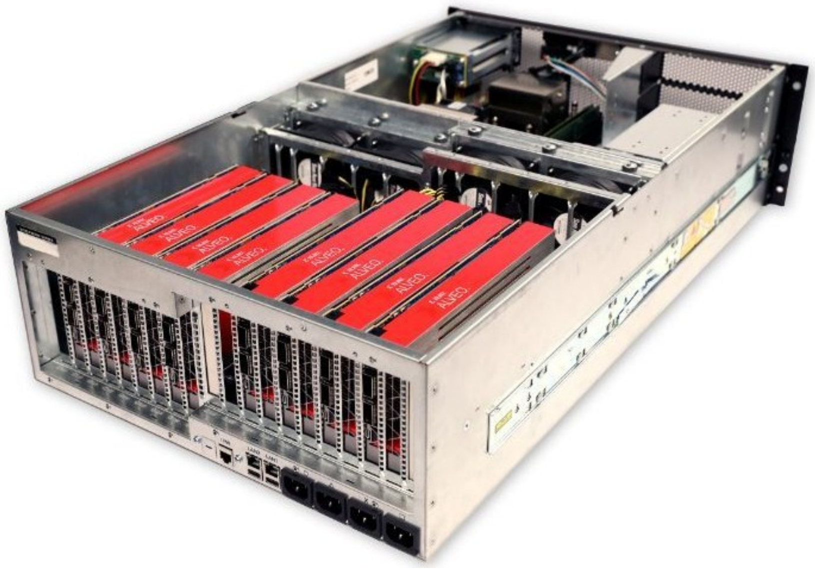
Goal ➝ Deploy FPGAs at <u>large scale</u> in hyperscale DCs

1-10s of thousands per DC

Requirements

- Server commodity & homogeneity
- Decrease in cos ower
- Easy to m deploy
- On
- migration
- S orchestration
- Hy & private
- Flex IaaS, PaaS, FaaS
- Clusters → #accelerators per server
- Community → #APPs, # developers

Scale Game
Fully driven by the
perf-per-cost metric



Agenda

- cloudFPGA – A bird's eye view
- Architecture & Design choices
  - HW: Boards, SLEDs, chassis
  - SW : Shell, Role, Management core
  - DC : Resource manager
- How-to cloudFPGA @ ZYC2
- Future work & Call for contribution

Part-1

cloudFPGA

A bird's eye view

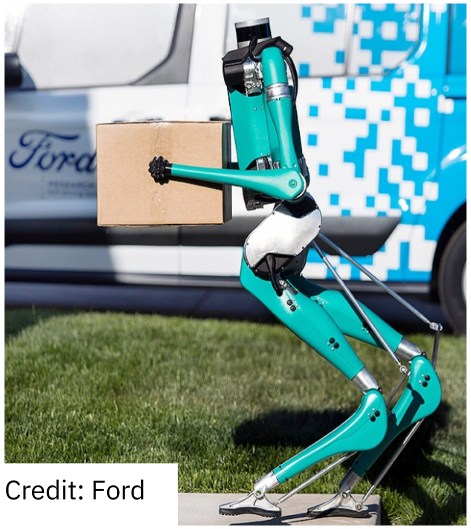
cloudFPGA in few words

- End of CPU slavery [3]
  - ↳ FPGAs becomes the compute node
- Standalone operation [4]
  - ↳ Disaggregated from CPU servers
  - ↳ Fast power-on / power-off
- Network-attached [5]
  - ↳ TCP-UDP / IP / Ethernet (currently 10-40GE)
  - ↳ Leaf-spine design
- Hyperscale infrastructure [6]
  - ↳ Focus on cost, energy, density, scalability
  - ↳ Promotes the use of mid-range FPGAs

cloudFPGA in few words

- End of CPU slavery [3]
  - ↳ FPGAs becomes the compute node
- Standalone operation [4]
  - ↳ Disaggregated from CPU servers
  - ↳ Fast power-on / power-off
- Network-attached [5]
  - ↳ TCP-UDP / IP / Ethernet (currently 10-40GE)
  - ↳ Leaf-spine design
- Hyperscale infrastructure [6]
  - ↳ Focus on cost, energy, density, scalability
  - ↳ Promotes the use of mid-range FPGAs

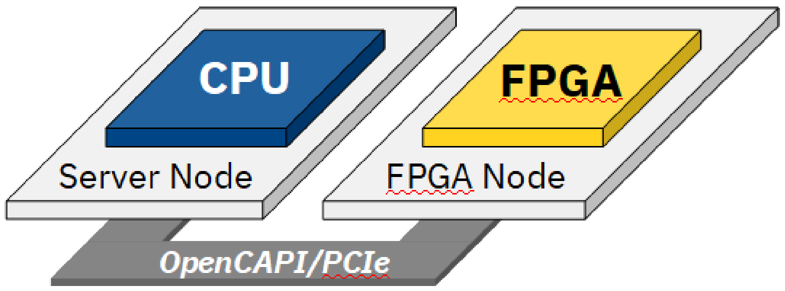
Credit: UPS


Credit: Ford

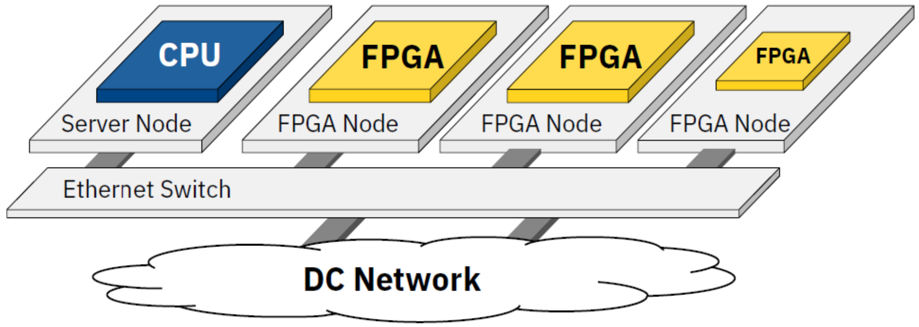
Credit: Amazon

FPGAs to become 1st class-citizens in DC-Cloud

FPGA as a Co-Processor

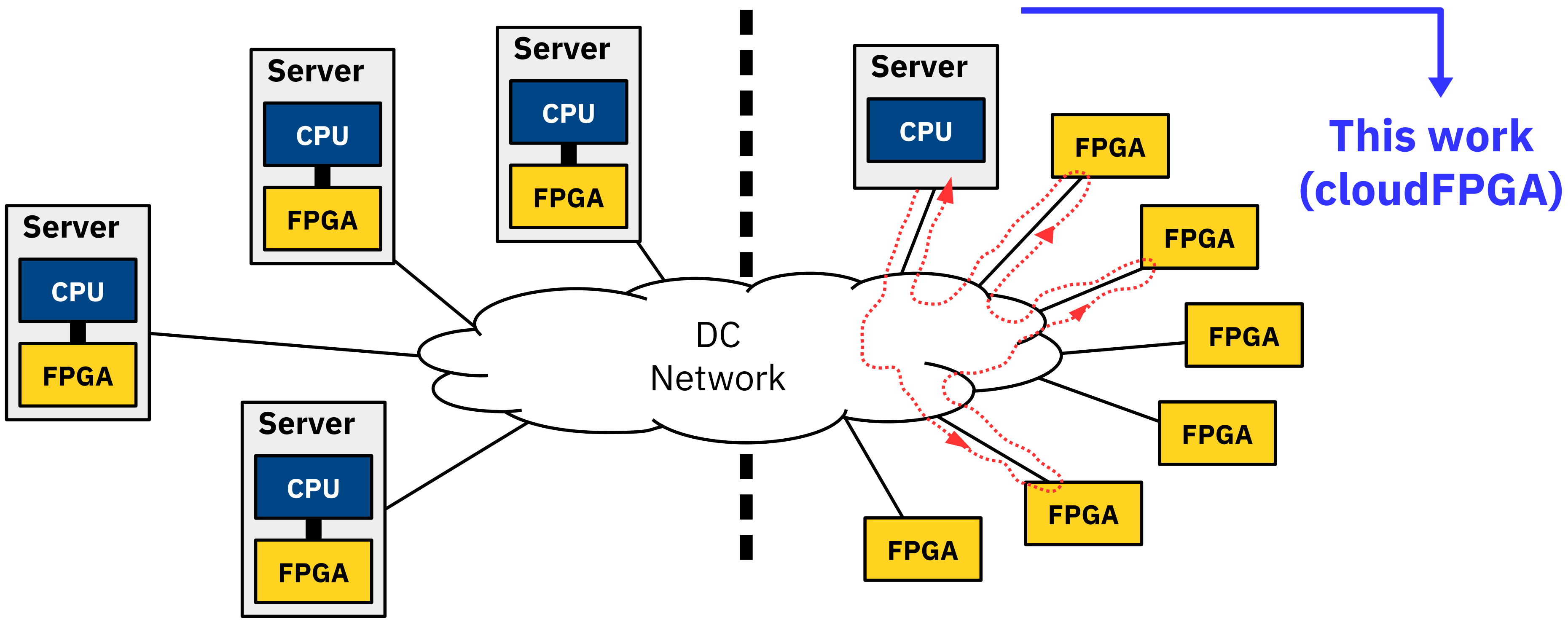


FPGA as a Peer-Processor



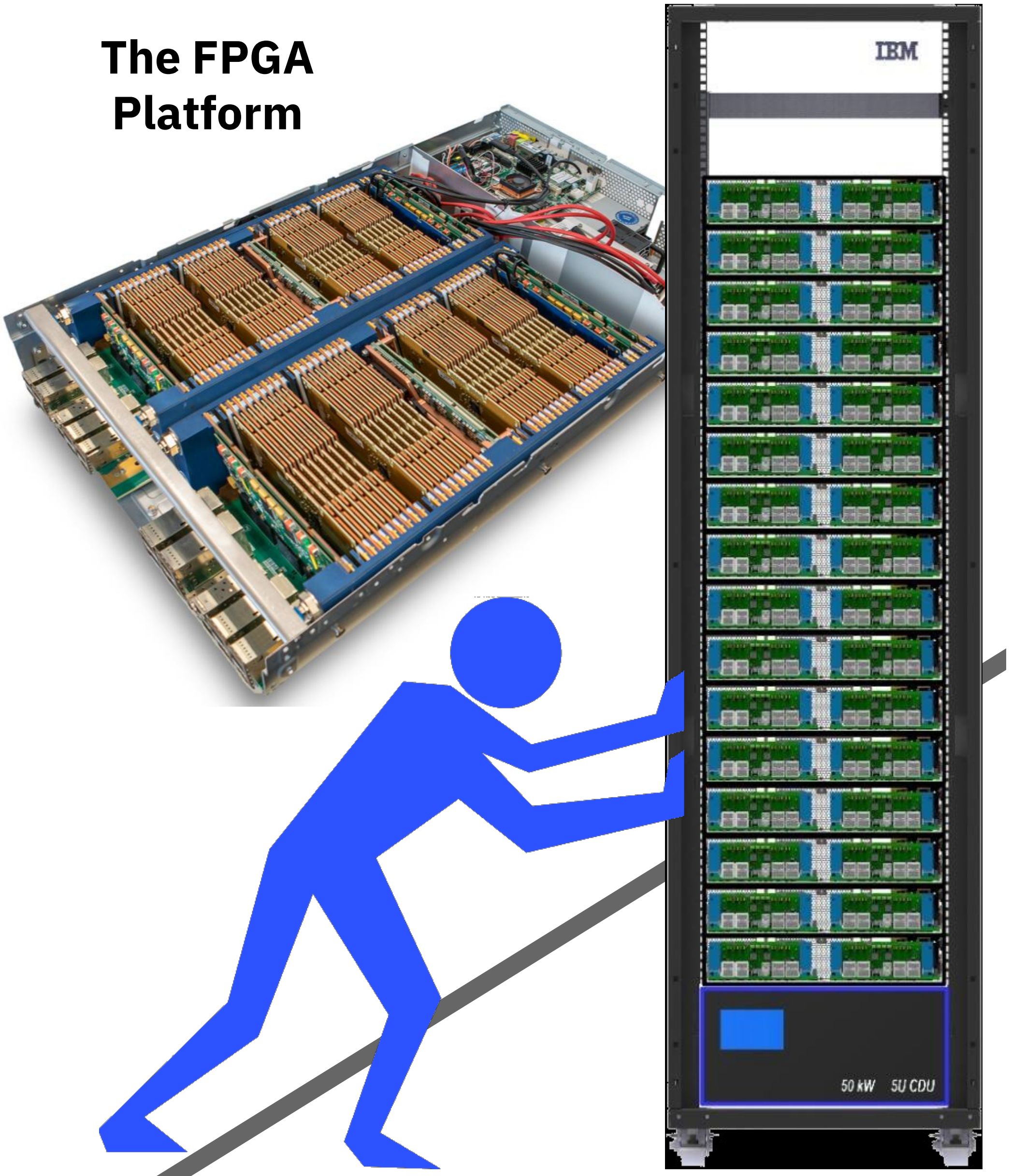
CPU-Centric Deployment

FPGA-Centric Deployment

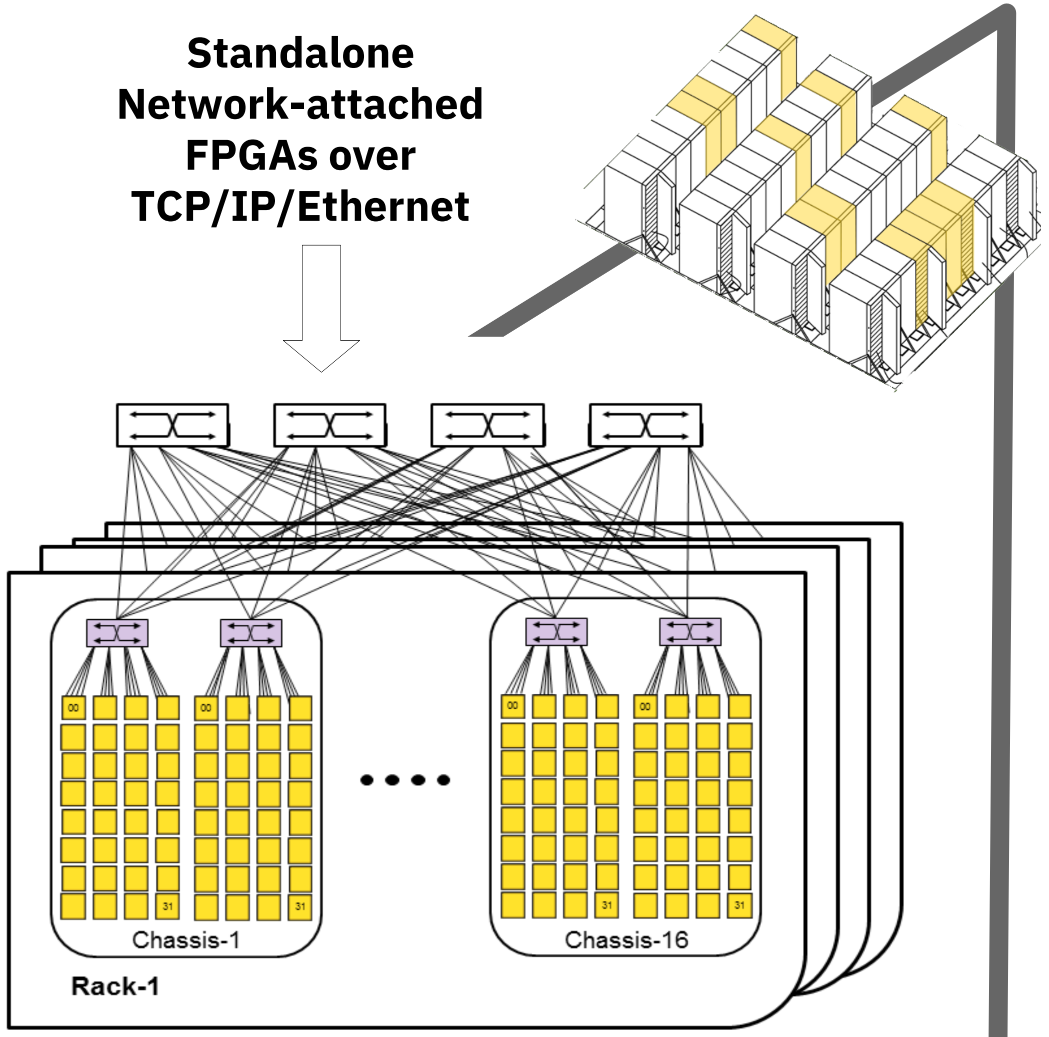


DC Vision = Hyperscale Infrastructure

The FPGA Platform



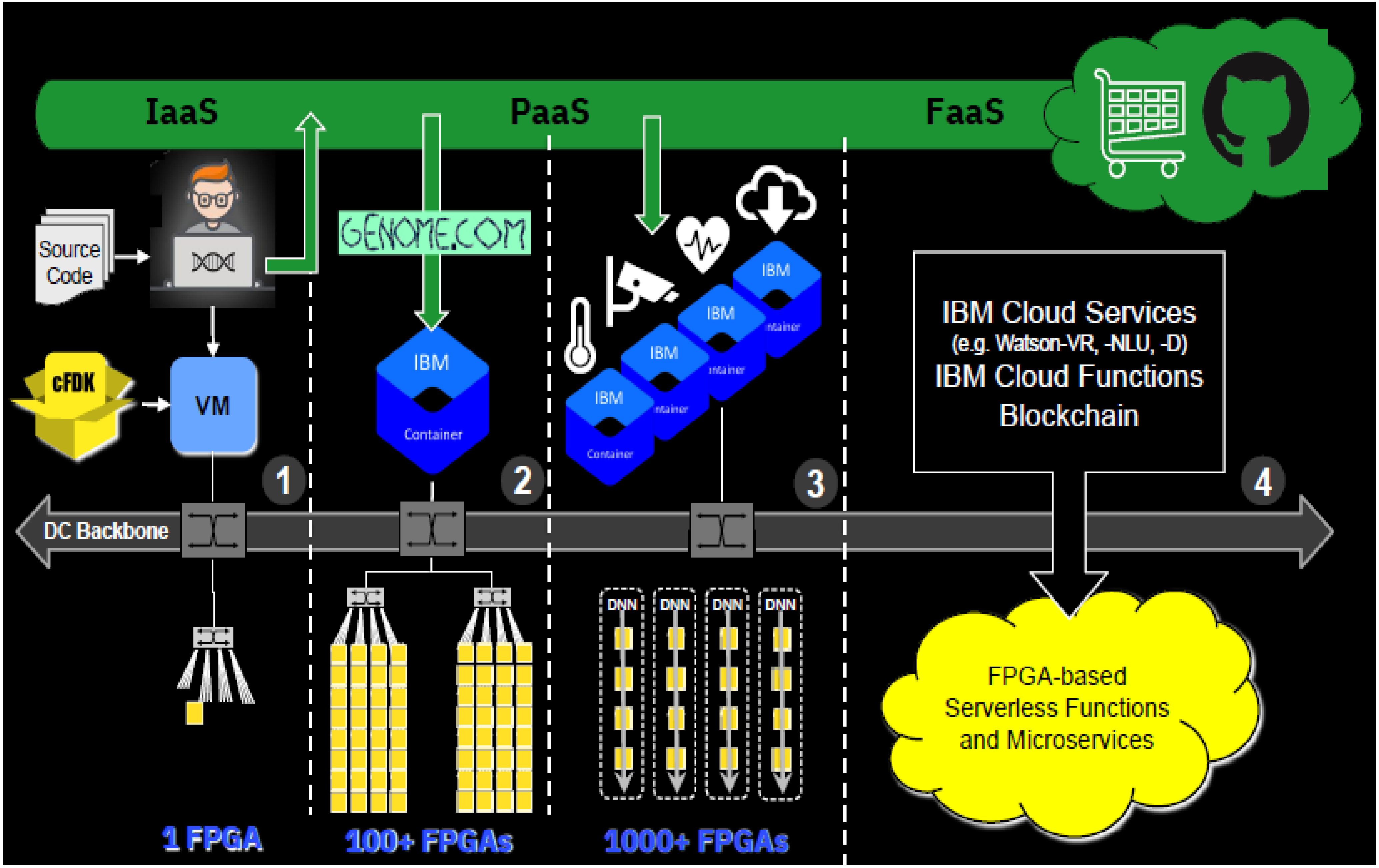
Standalone Network-attached FPGAs over TCP/IP/Ethernet



10 Tb/s full-duplex

64/chassis — 1024/rack — Plentiful/DC

Cloud Vision = IaaS, PaaS, FaaS



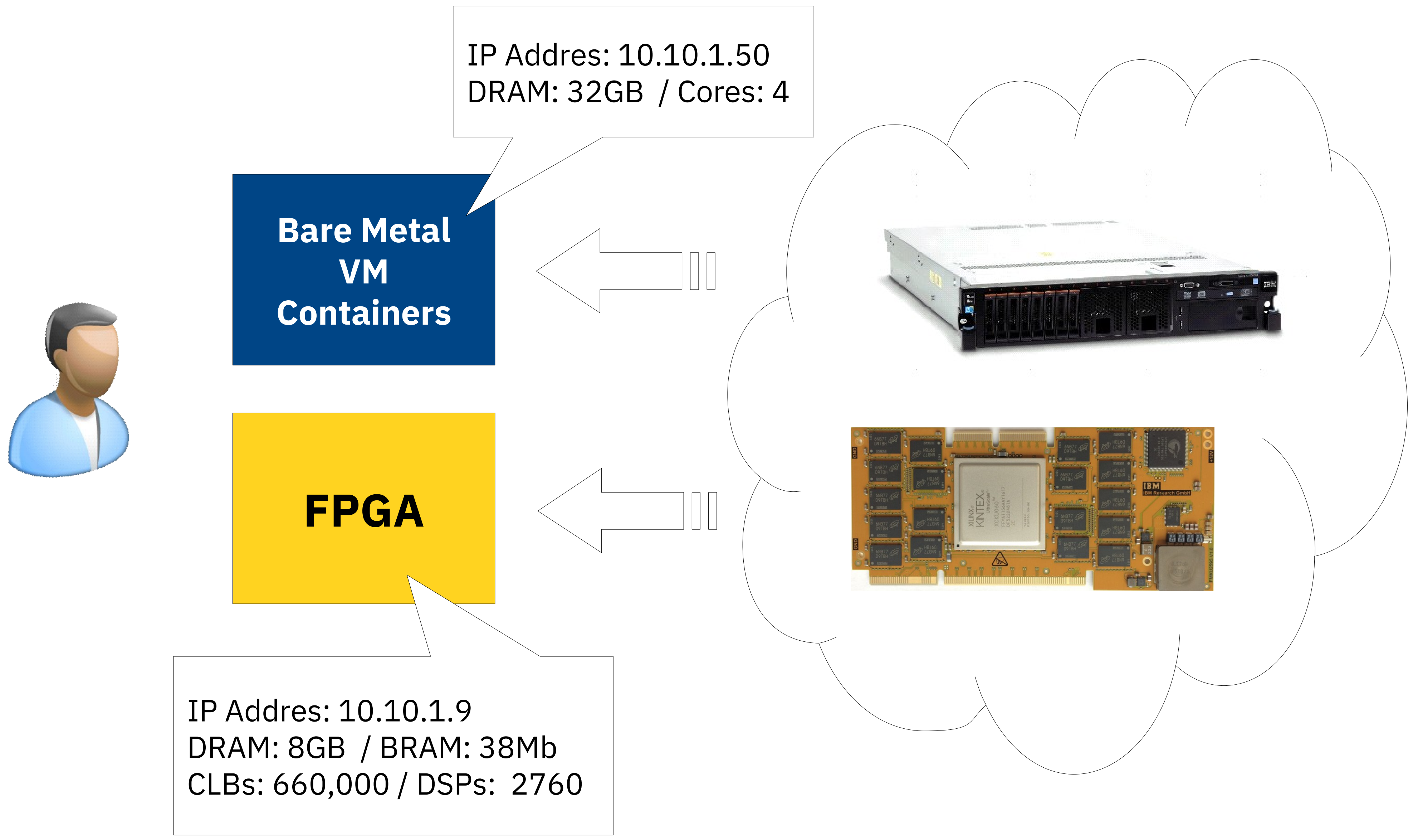
Part-2a

Architecture & Design choices

HW: Boards, SLEDs, chassis

Standalone → The FPGA becomes the node

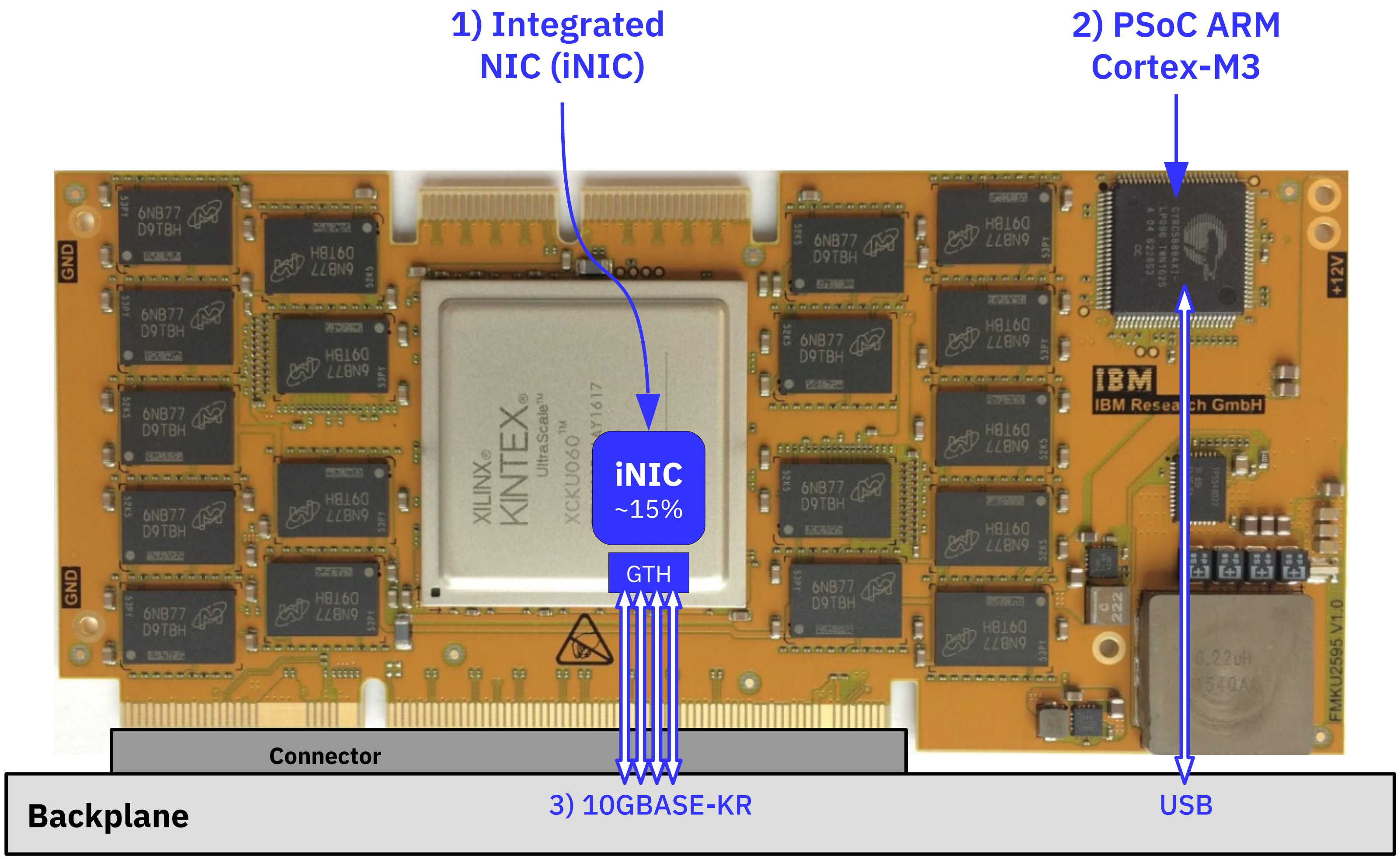
- Disaggregation of the FPGA from the server



IP Addres: 10.10.1.50
DRAM: 32GB / Cores: 4

IP Addres: 10.10.1.9
DRAM: 8GB / BRAM: 38Mb
CLBs: 660,000 / DSPs: 2760

Standalone network-attached FPGA



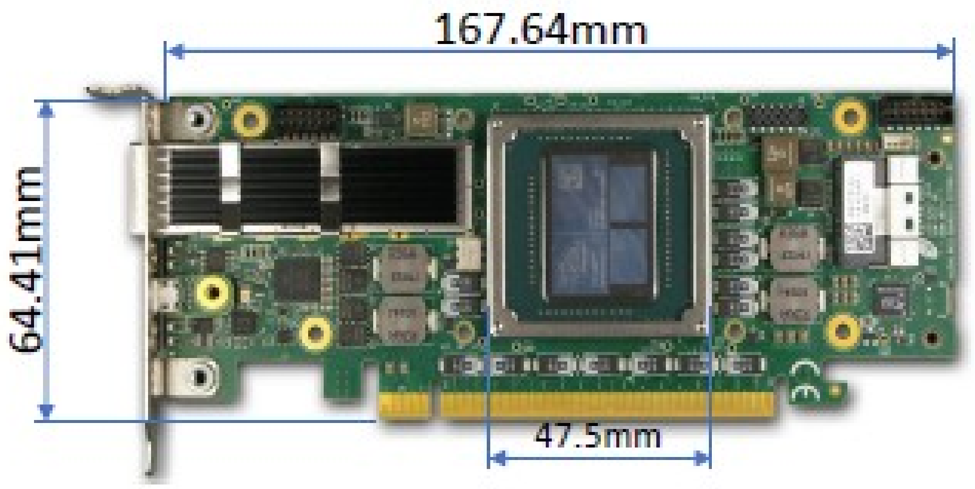
1) Replace PCIe I/F with integrated NIC (iNIC)

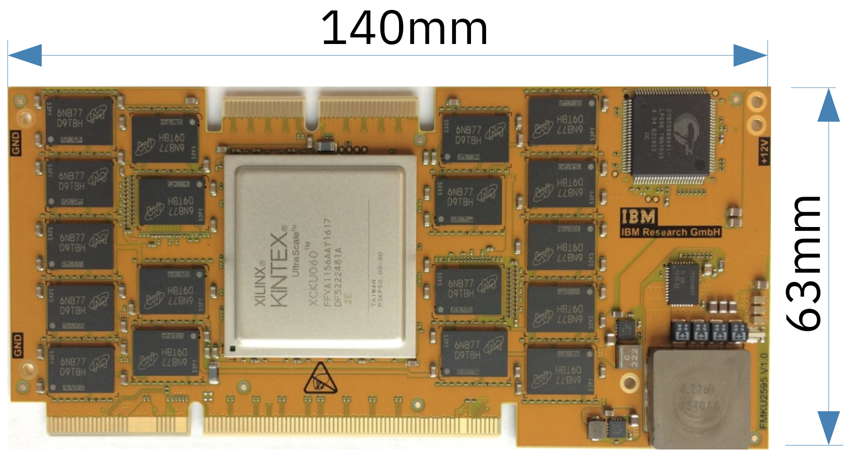
2) Turn FPGA card into a standalone resource

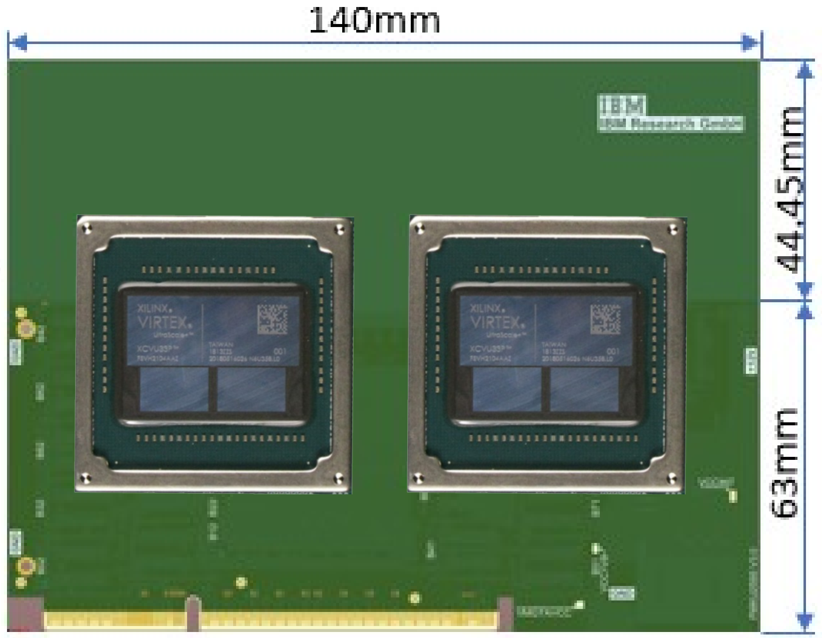
3) Replace transceivers w/ backplane connectivity

How does it compare w/ PCIe cards?

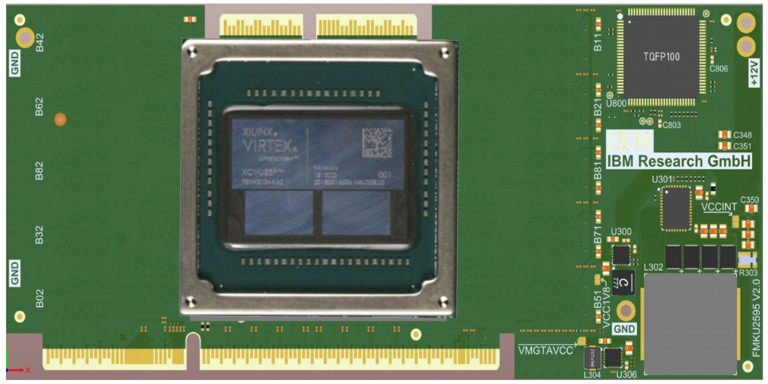
For comparison: ALPHA DATA ADM-PCIE-9H3,
1/2 Length, low profile, x16 PCIe form Factor





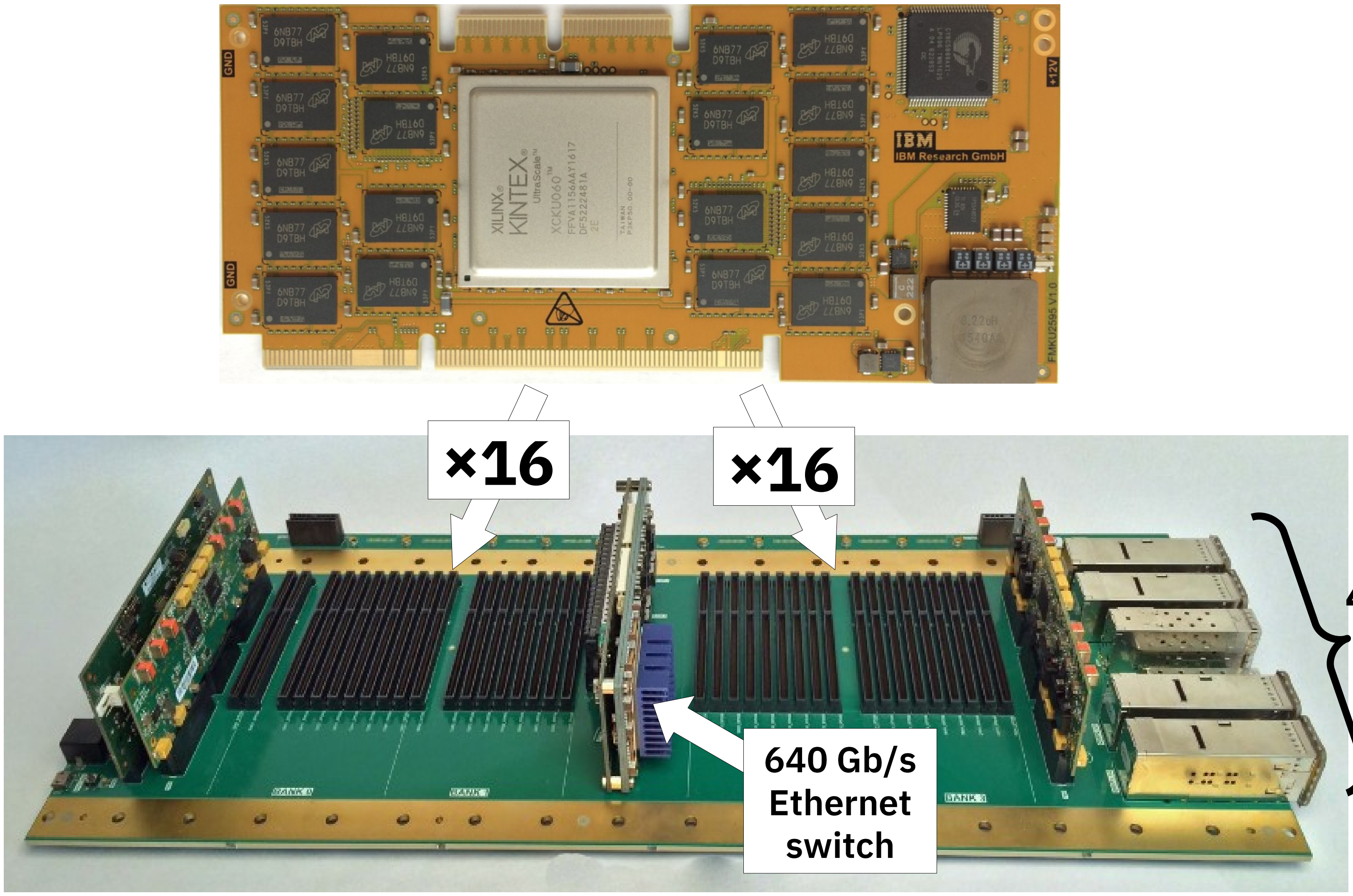


Figurative picture

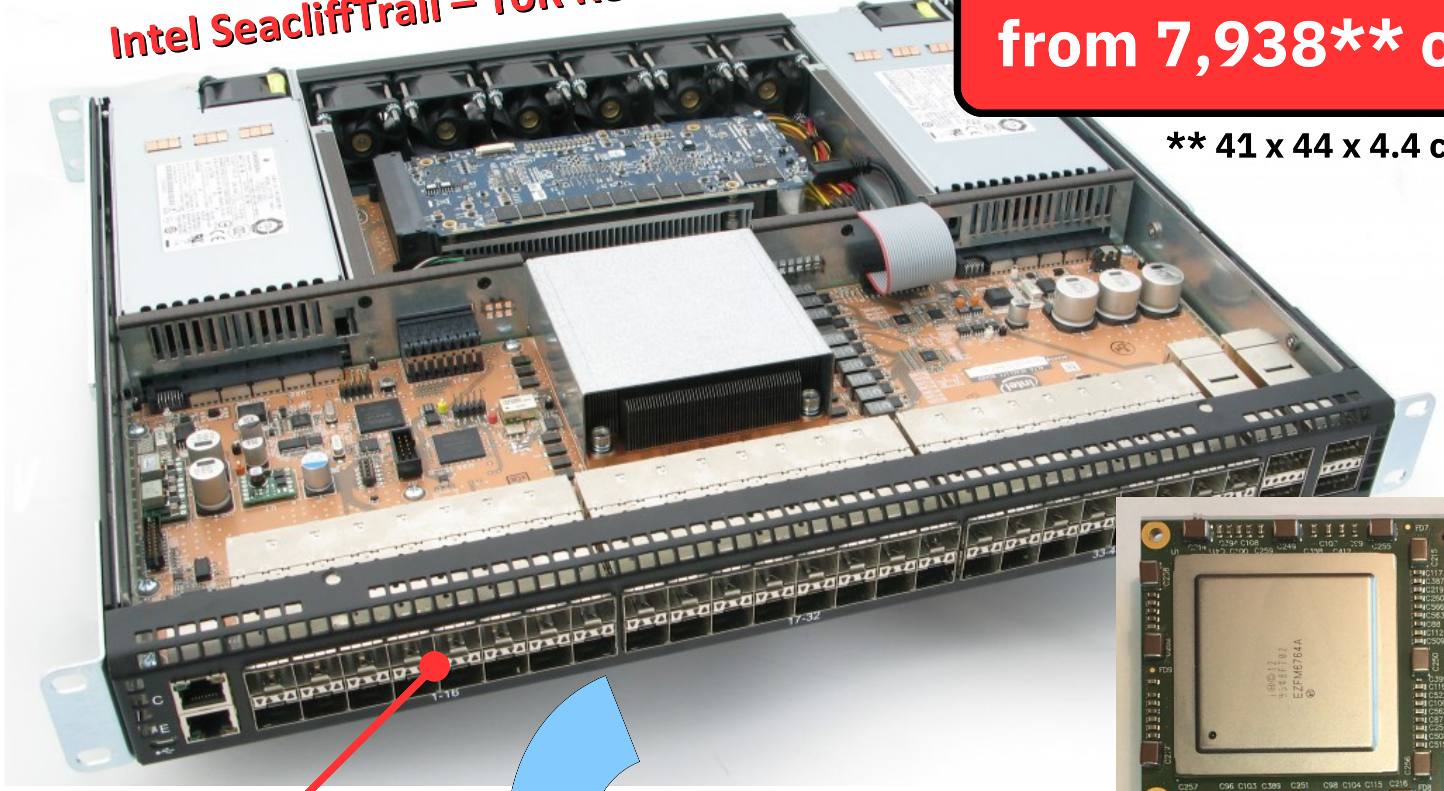


Figurative picture

One carrier SLED (a.k.a PoD)= 32 FPGA modules



From top-of-rack down to SLED/PoD switch



from 7,938 cm^3**

** 41 x 44 x 4.4 cm

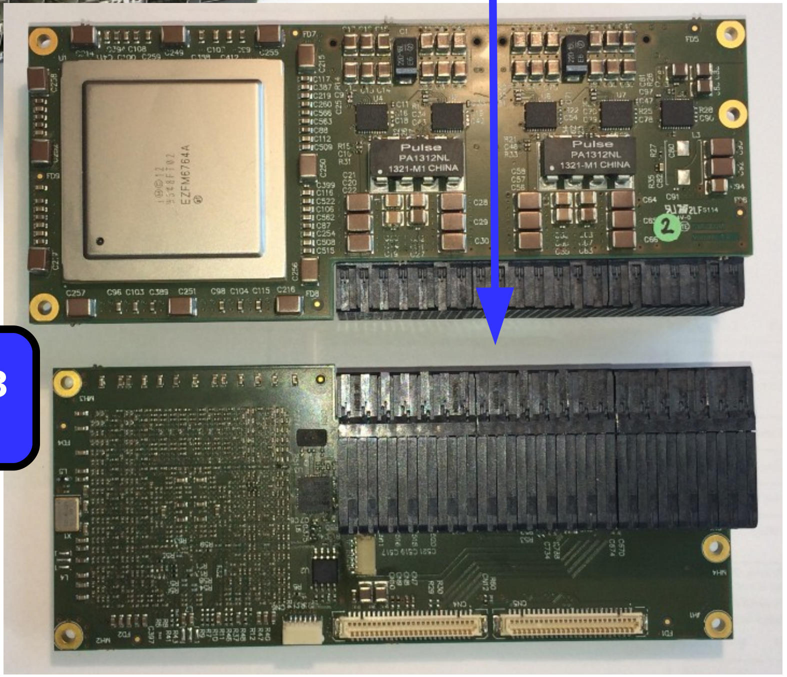
48 x 10 GbE + 4 x 40 GbE

1/21

to 378* cm^3

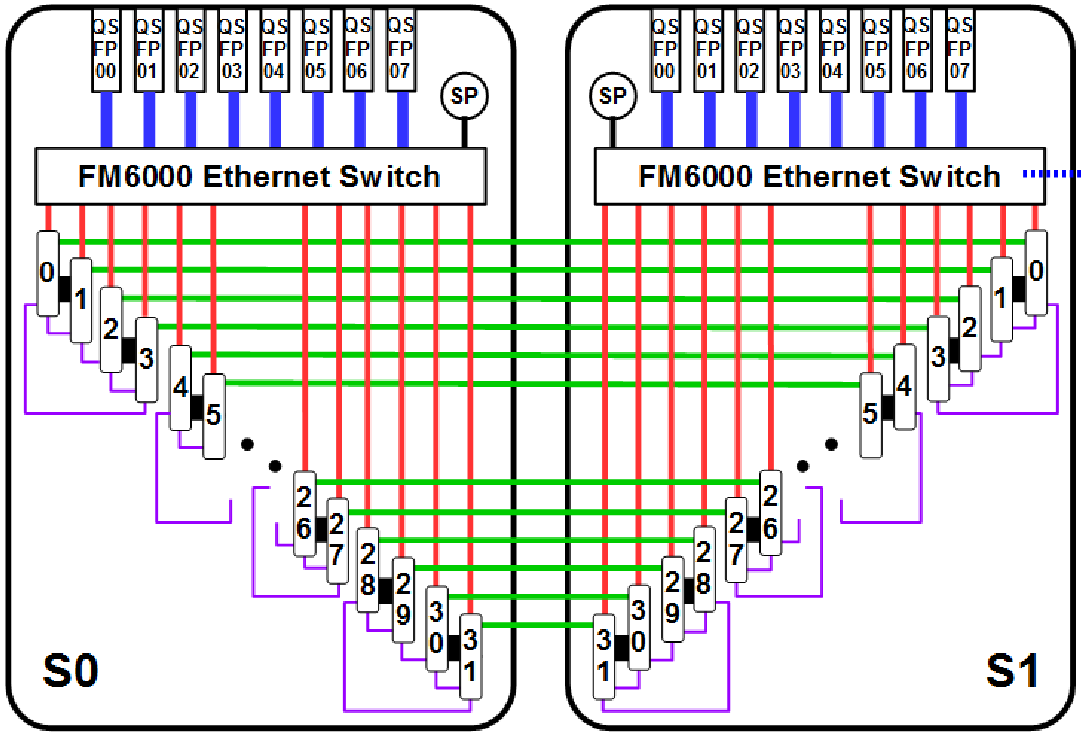
* 14 x 6 x 4.5 cm

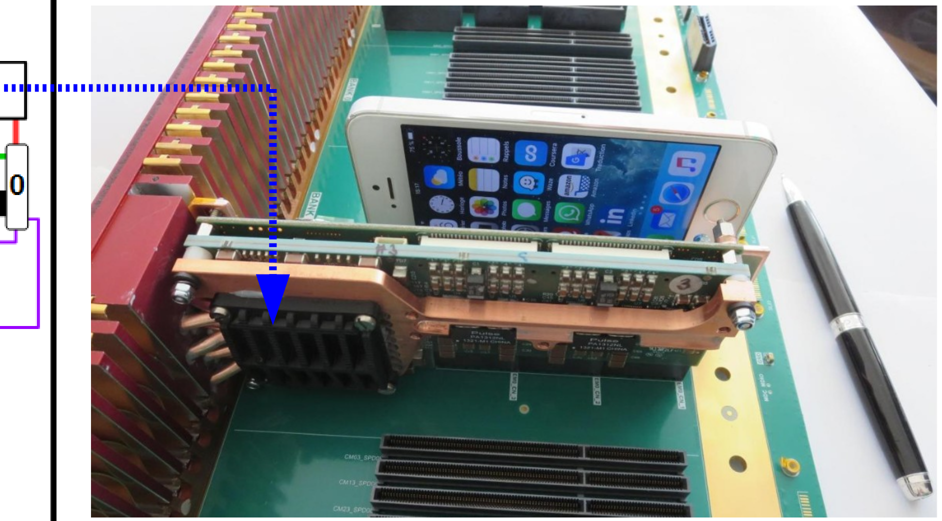
32x10 GbE + 8x40 GbE



Switch Module SM6000

Network topology per chassis = 64 FPGAs



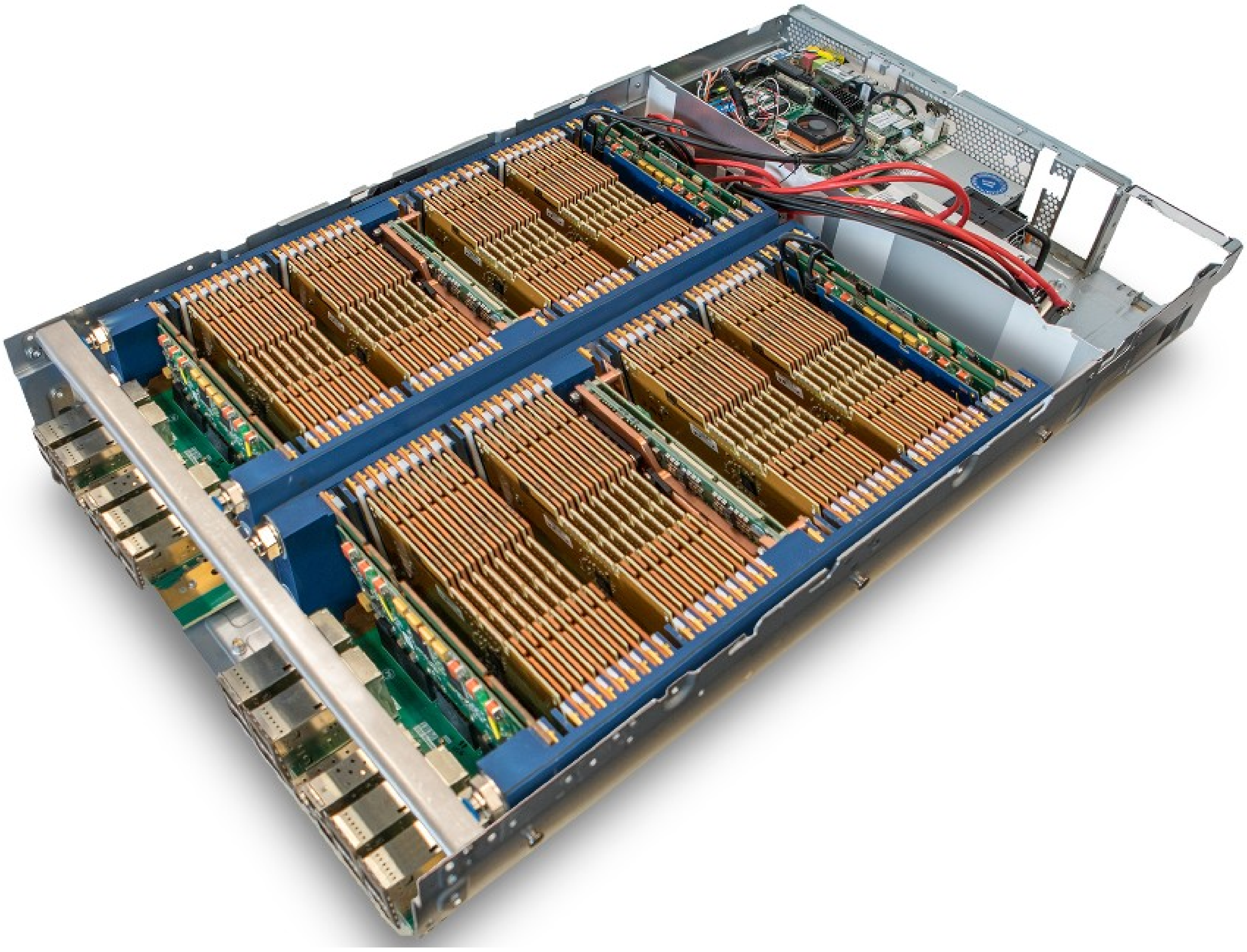


Integration size scale:
Ethernet switch FM6000 (64x10GbE) vs iPhone5

Legend (per slice):

[==]	x8 40GbE up links	(320 Gb/s)	Balanced (i.e. no over-subscription) between the north and south links of the Ethernet switch
[--]	x32 10GbE FPGA-to-Switch links	(320 Gb/s)	
[--]	x32 10GbE redundant links		
[--]	x32 10GbE FPGA-to-FPGA links		
[■]	x16 PCIe x8 Gen3		
SP	x1 Service Processor		

The cloudFPGA Platform (19”x2U w/64 FPGAs)

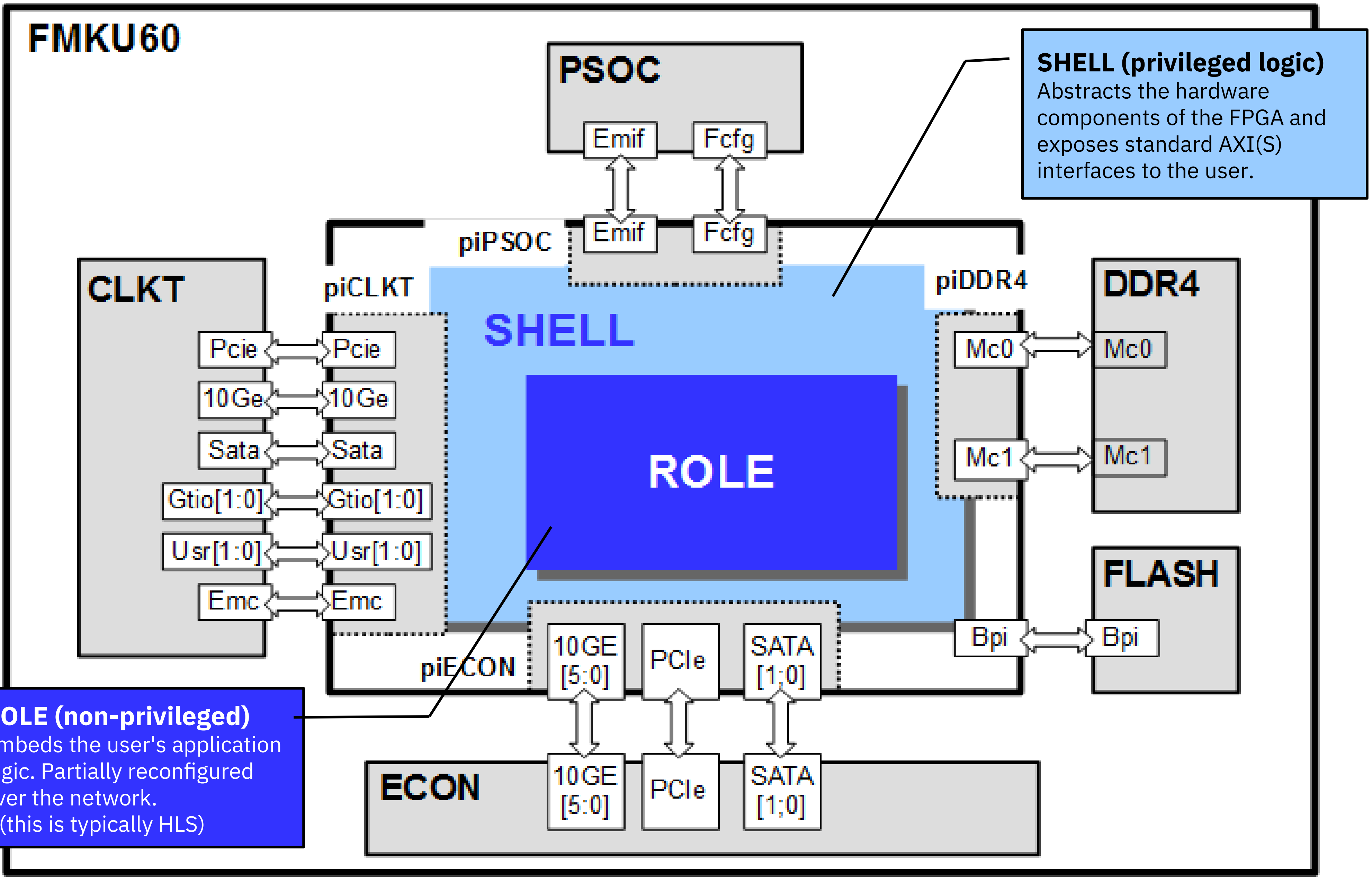


Part-2b

Architecture & Design choices

SW: Shell, Role, Mngt core

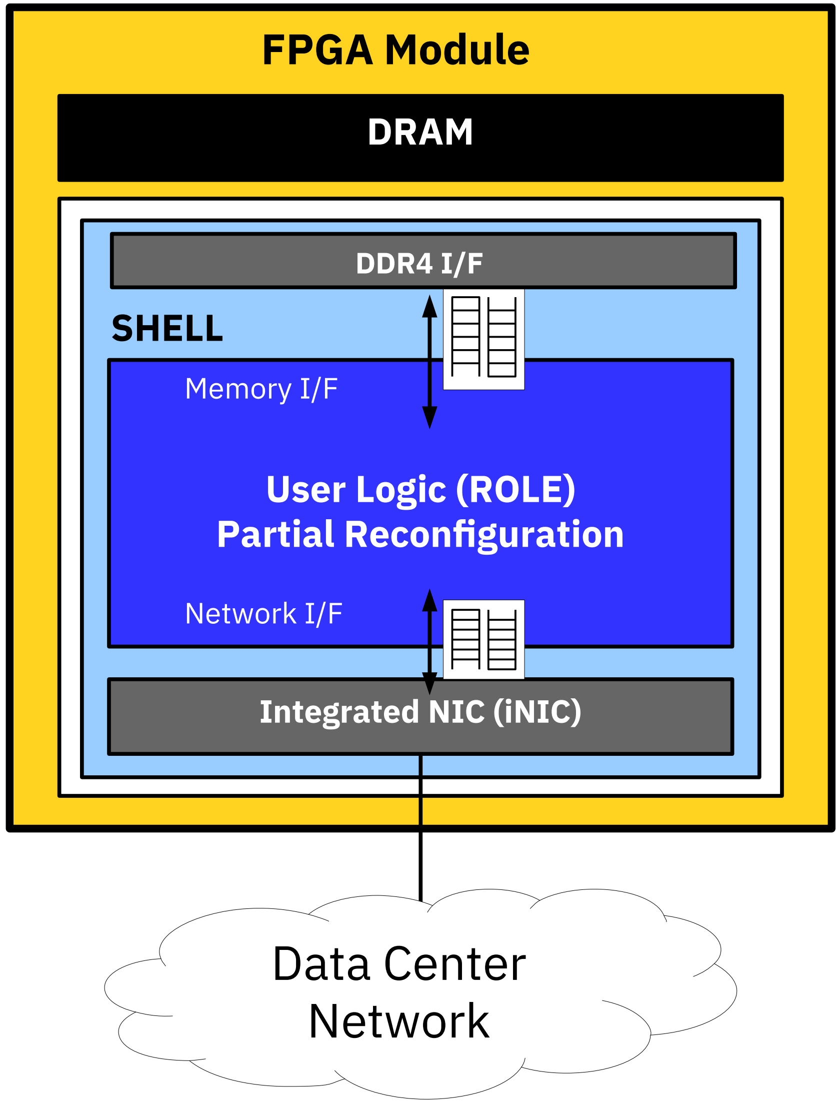
Hw Abstraction → Shell Role Architecture (SRA)

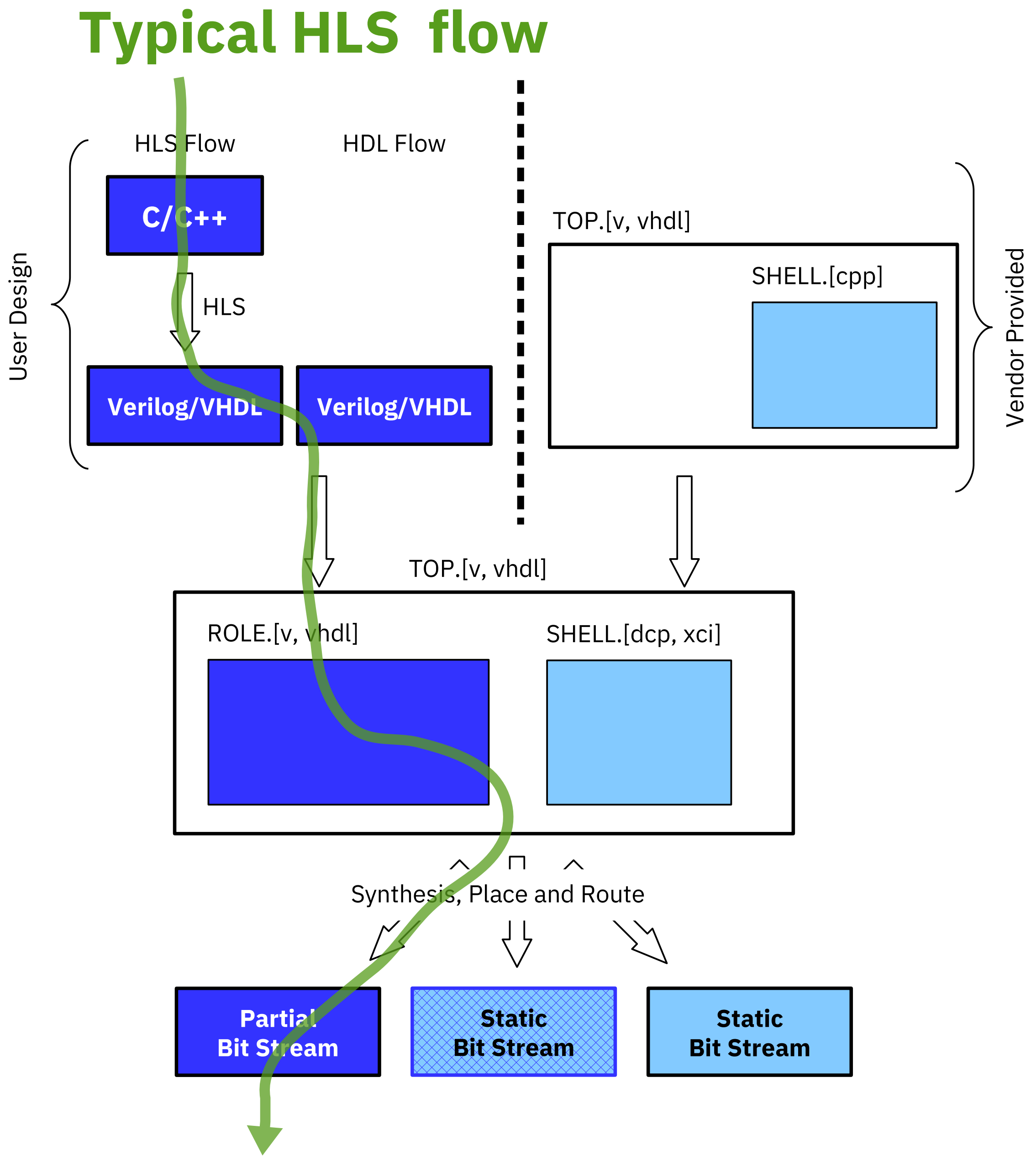


SHELL (privileged logic)
Abstracts the hardware components of the FPGA and exposes standard AXI(S) interfaces to the user.

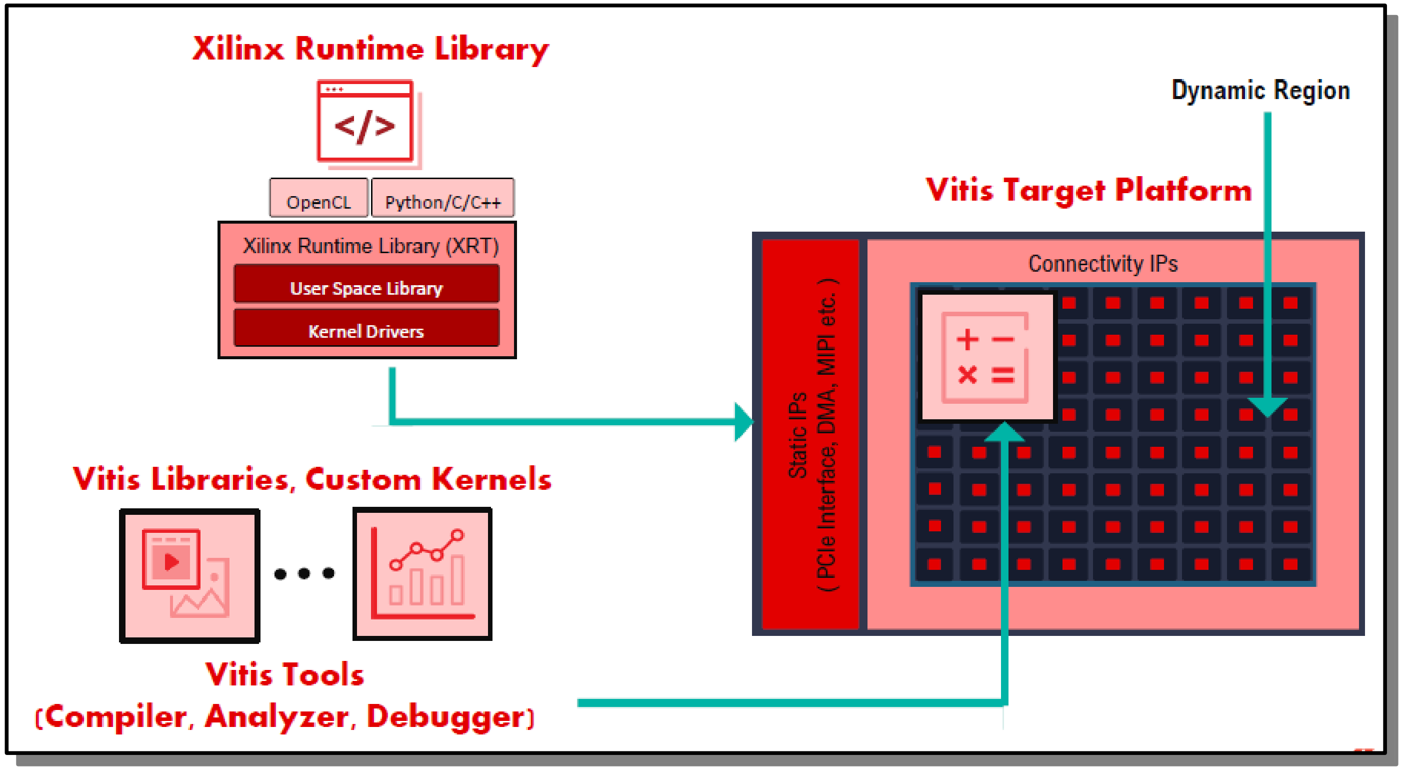
ROLE (non-privileged)
Embeds the user's application logic. Partially reconfigured over the network.
(this is typically HLS)

cloudFPGA Development Kit (cFDK)





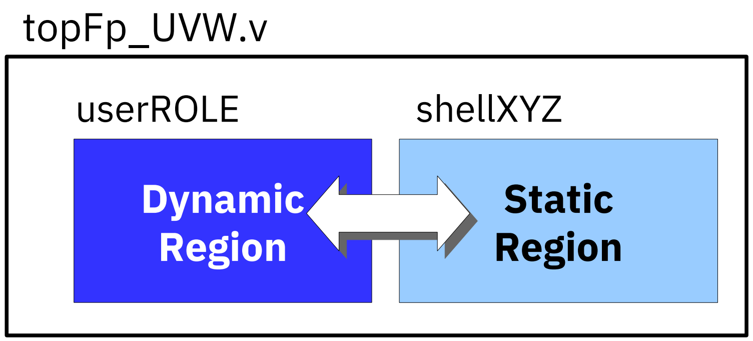
Analogy with Vitis target platforms



Source: Development with Vitis Accelerated Libraries, Xilinx 2020

cF-SHELL → Static Region → Privileged logic → Configured after power-on via local Flash.

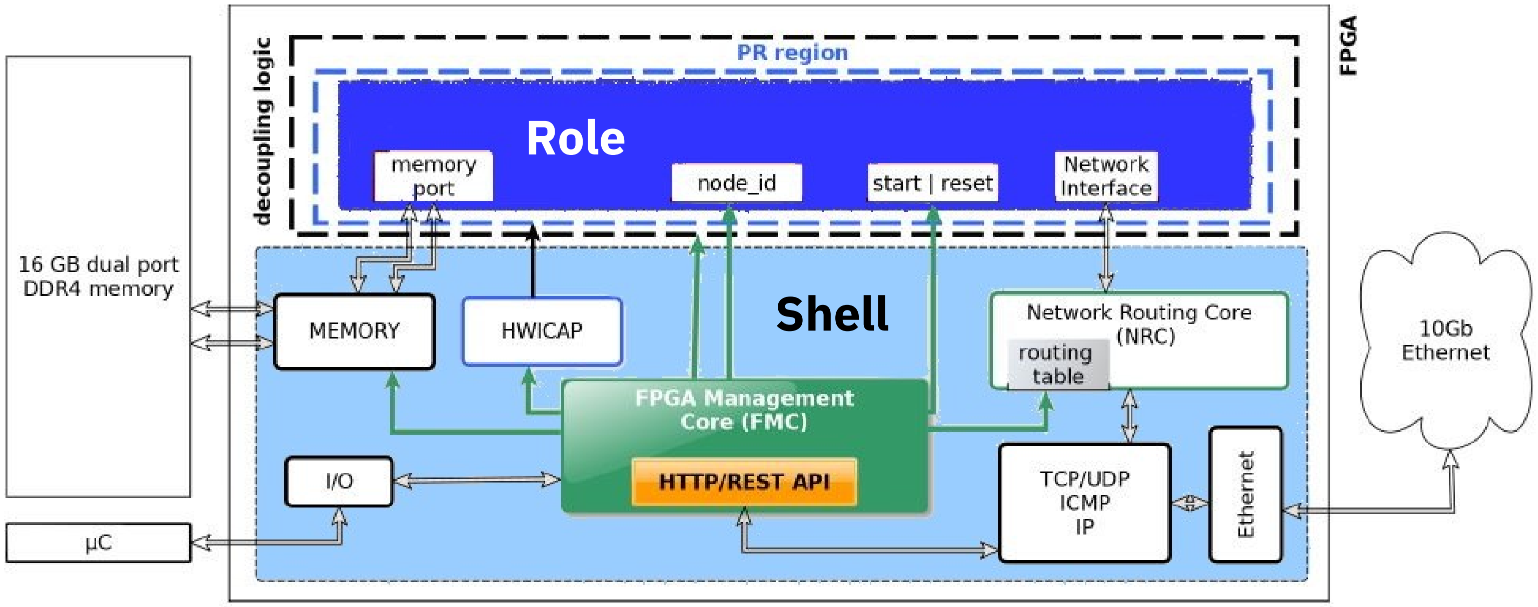
cF-ROLE → Dynamic Region → Non-privileged logic → Partially configured over the network.



FPGA Management Core

There is one management core per FPGA (FMC)

- The FMC contains a simplified HTTP server which provides support for the REST API calls issued by the Data Center Resource Manager (DCRM) [7].



The FMC understands REST API calls such as:

• `POST /configure`	Submits a partial bitfile and triggers the PR of the Role region.
• `GET /status`	Returns some application-specific status information.
• `PUT /node_id`	Sets the node-id register of the Role.
• `POST /routing`	Sends the routing information of a cluster to the FPGA.

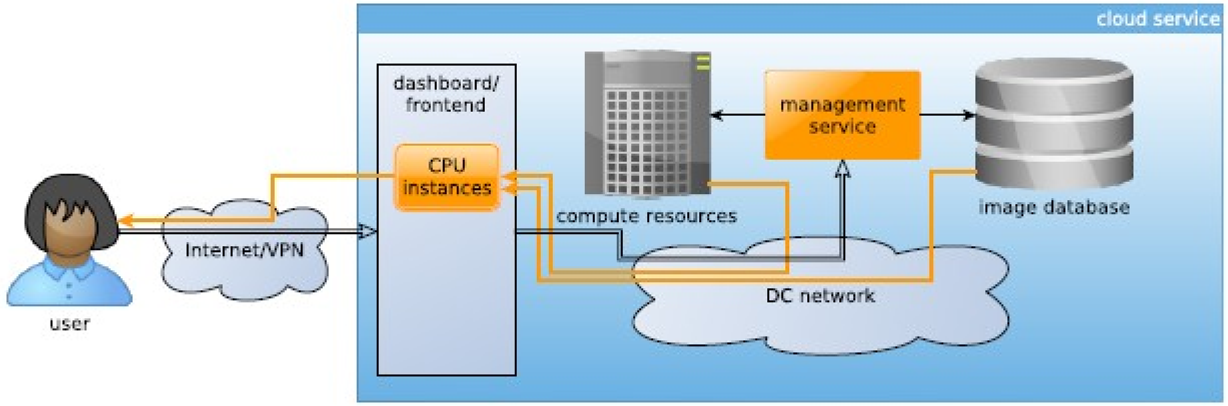
Part-2c

Architecture & Design choices

DC: Resource manager

Cloud Service Architecture for FPGAs (1/2)

- **Instance** = Resource + Image
- **Cluster** = *N* * *Instance*

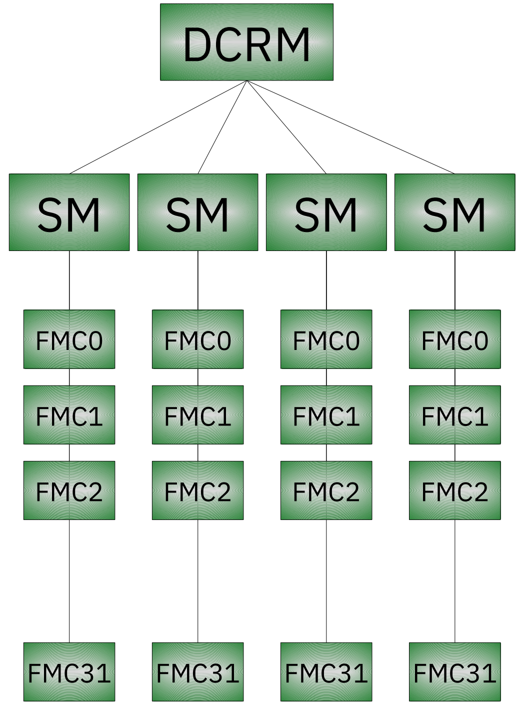


A typical cloud service hosting VMs has three components:

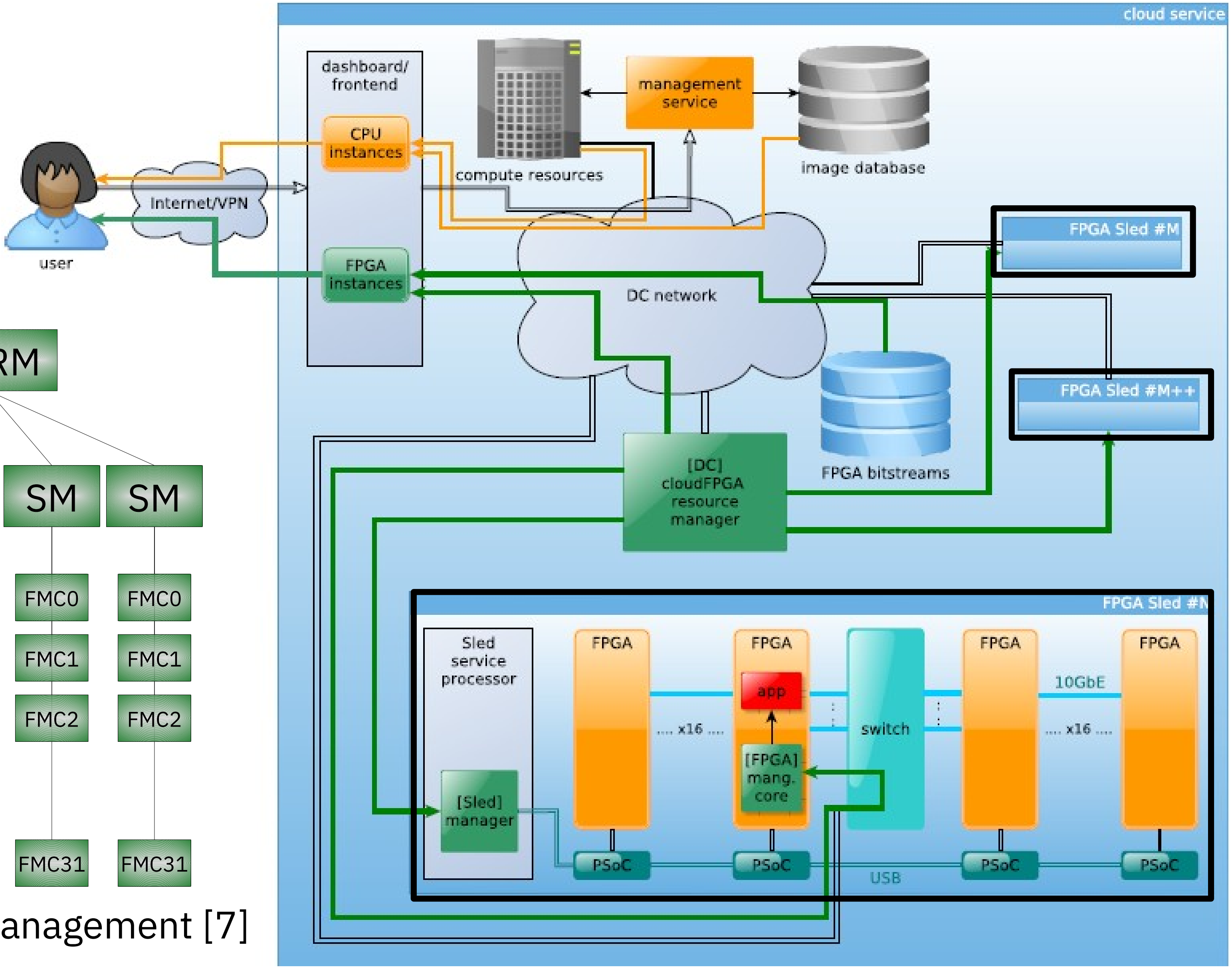
- A pool of compute resources
- A database of VM images
- A management service

Cloud Service Architecture for FPGAs (2/2)

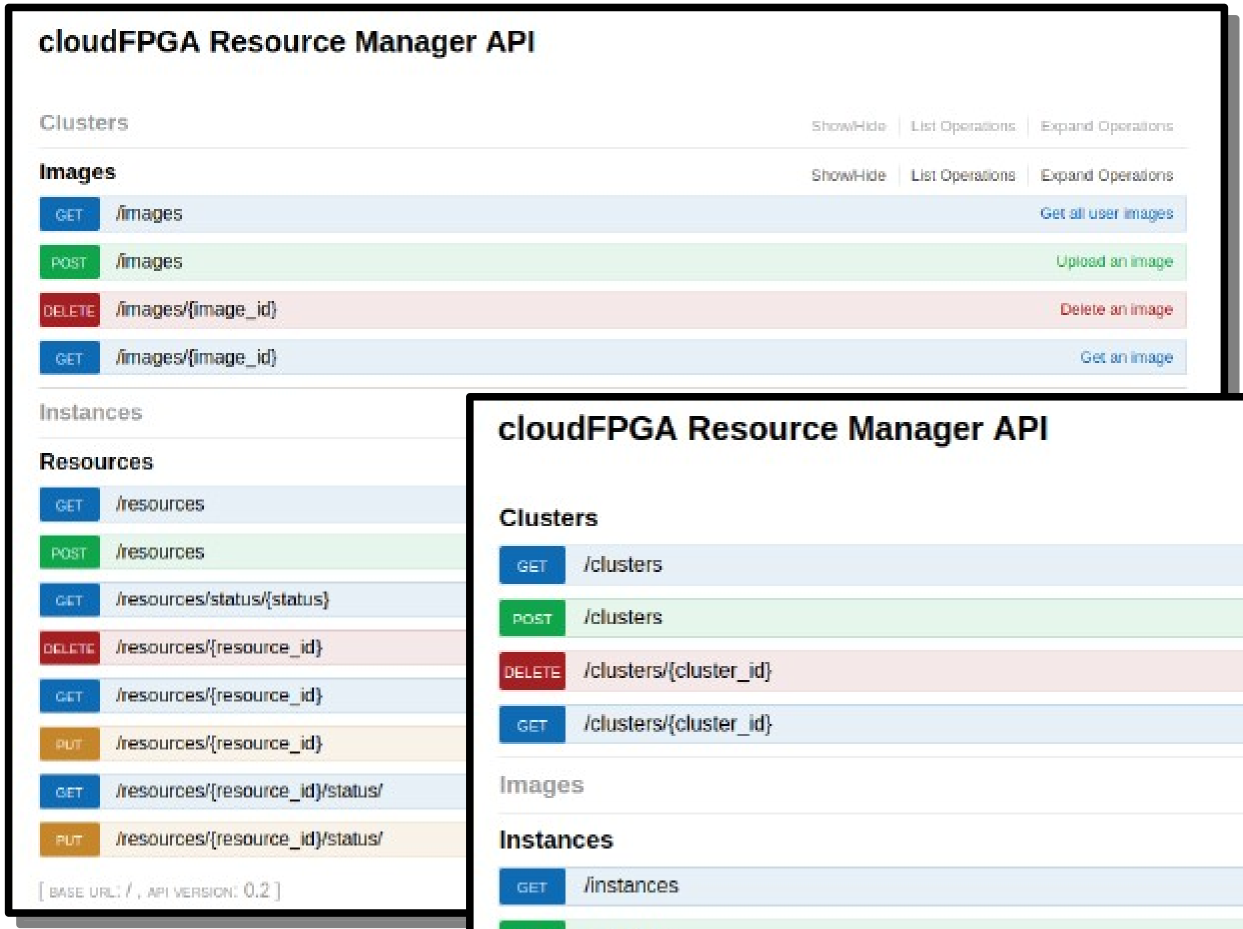
↳ Instance = FPGA + Bitstream



Hierarchical Management [7]



RESTful Web API Based



cloudFPGA Resource Manager API

Clusters — Show/Hide | List Operations | Expand Operations

Images — Show/Hide | List Operations | Expand Operations

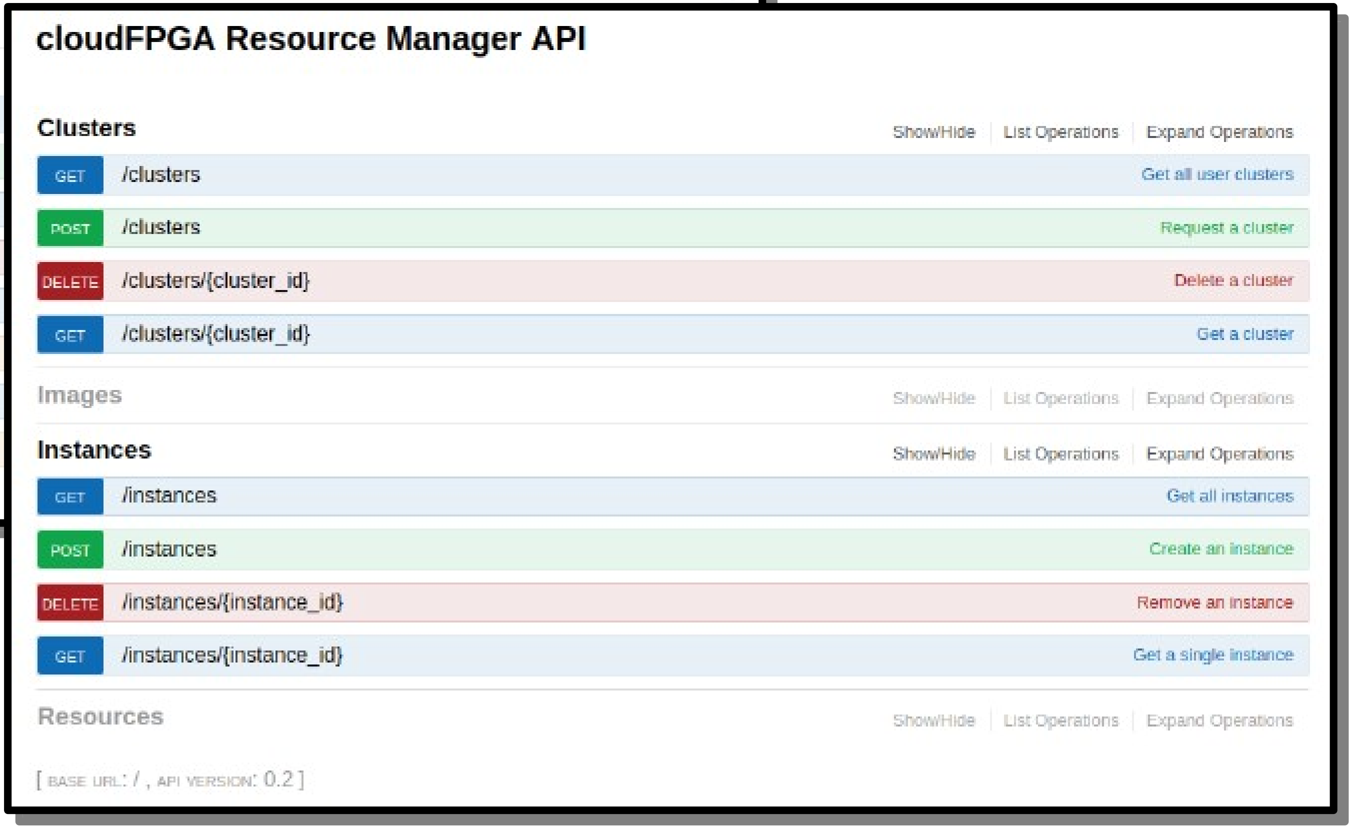
GET	/images	Get all user images
POST	/images	Upload an image
DELETE	/images/{image_id}	Delete an image
GET	/images/{image_id}	Get an image

Instances

Resources

GET	/resources
POST	/resources
GET	/resources/status/{status}
DELETE	/resources/{resource_id}
GET	/resources/{resource_id}
PUT	/resources/{resource_id}
GET	/resources/{resource_id}/status/
PUT	/resources/{resource_id}/status/

[BASE URL: / , API VERSION: 0.2]



cloudFPGA Resource Manager API

Clusters — Show/Hide | List Operations | Expand Operations

GET	/clusters	Get all user clusters
POST	/clusters	Request a cluster
DELETE	/clusters/{cluster_id}	Delete a cluster
GET	/clusters/{cluster_id}	Get a cluster

Images — Show/Hide | List Operations | Expand Operations

Instances — Show/Hide | List Operations | Expand Operations

GET	/instances	Get all instances
POST	/instances	Create an instance
DELETE	/instances/{instance_id}	Remove an instance
GET	/instances/{instance_id}	Get a single instance

Resources — Show/Hide | List Operations | Expand Operations

[BASE URL: / , API VERSION: 0.2]

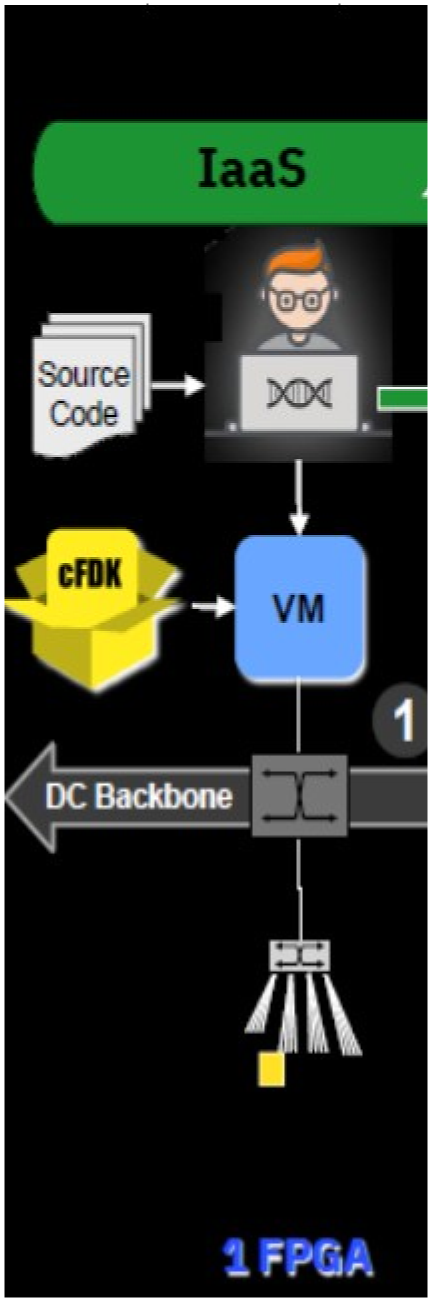
Part-3

How-to

cloudFPGA @ ZYC2*

*On-premise deployment in the Zurich Yellow security zone Compute Cluster

Example #1 – "Hello, World!" with a single FPGA



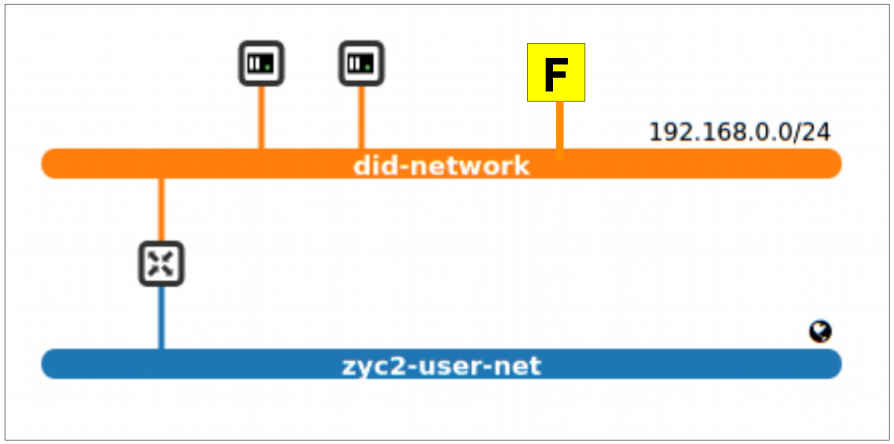
1) Download the cFDK to work remotely on your desktop or use a VM @ ZYC2

2) Setup a VPN client, create an OpenStack project and a private network for it

3) Develop and simulate

4) Place and route

5) Upload your bitstream
   – You'll receive an *image-id*

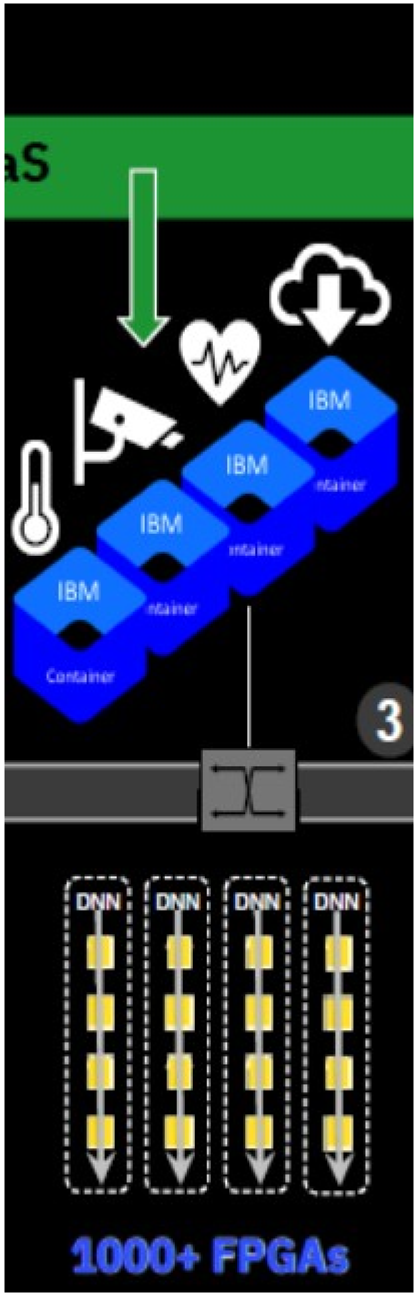


6) Request an instance to be launched with your *image-id*
   – You'll get back an *image-IP* and an *instance-id*

7) Ping the *image-IP*

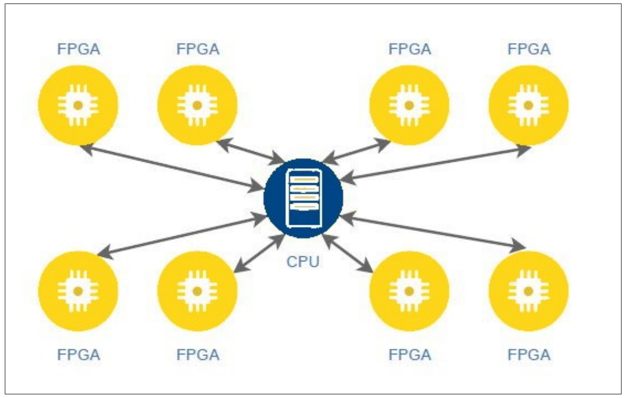
8) You are ready to communicate with your FPGA via network sockets with TCP or UDP protocol

Example #2 – "Stencil comp." → 1 HOST + 8 FPGAs



1) Design your FPGA kernel(s) and your HOST code

```
while ((not converged) and (max iterations not reached)) {
for (i,j) distribute over all nodes {
  if(i is border or j is border)
    continue;
  xnew[i][j] = (x[i+1][j] + x[i-1][j] + x[i][j+1] + x[i][j
-1])/4;
}
for (i,j)       # done on one node
  x[i][j] = xnew[i][j];
}
```



2) Build, place and route

3) Use a script to interact with the RESTfull Web API
- Upload the bitstream(s)
- Request a cluster to be launched
- Let the HOST send and receive to/from the FPGAs

```
$> myStencilComp new a202ad9e-0981-4c1b-b2fe-8879354bbb777 8 ./myServerCode
```

Request a new cluster (or re-use a cluster #) | Image Id | #FPGAs | SW binary

FYI, see also [8]: A one-click solution which compiles a standard MPI application for a Reconfigurable Heterogeneous Computing Cluster (ReH^{2}PC).

Part-4

Future work

Future Work (1/2)

1) Open-source the cloudFPGA Development Kit (cFDK)

– Coming soon
– Give the research community access to cloudFPGA platform

2) Walking up the application stack

– Lower-precision inference and autoML
– Support for Vitis accelerated libraries
– Large-scale distributed applications
– Support popular programming languages and frameworks

3) Walking up the systems stack

– Function-as-a-Service (aka Serverless computing)
– Composable and disaggregated storage (NVMe-oF)
– Lighter and faster network protocols

Future Work (2/2)

4) Expand the numbers of Xilinx-based modules

– Produce a stronger FPGA module (e.g. XCVU33P w/ HBM)

– Produce a module with an MPSoC or a Versal ACAP

5) Support faster link technologies

– 25GE, 100GE

6) Support other FPGA vendors

– Intel, Achronix, ...

7) Share the cloudFPGA platform design (e.g. à la OCP)

The cloudFPGA suspects

Current team

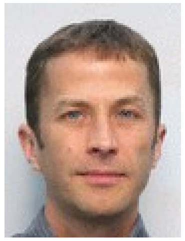
Beat

Burkhard

Christoph

Francois

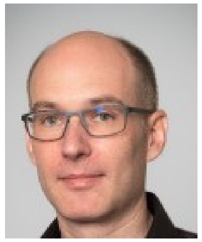
Dionysios


Mark

Mitra

Raphael

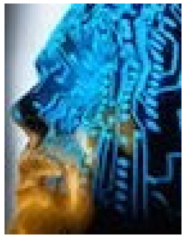
Former contributors


Alex

Andreas

Giorgio

Jagath


Ron

Stephan

Summary

1) FPGAs are eligible to become 1st class citizens
   - Standalone approach sets the FPGA free from the CPU
     - Large scale deployment of FPGAs independent of #servers
     - Significantly lowers the entry barrier
   - Promotes the use of medium and low-cost FPGAs

2) The network-attachment model
   - Makes FPGAs IP-addressable and scalable in DCs
     - Users can rent and link them in any type of topology
   - Opens the path for use of FPGAs in large scale applications
     - Serverless computing, HPC, DNN inference, Signal Processing, ...

3) The hyperscale infrastructure
   - Integrates FPGAs at the chassis (aka drawer) level
   - Combines passive and active water cooling
   - Key enabler for FPGAs to become plentiful in DCs

Thank you

May the FPGA be with you

https://www.zurich.ibm.com/cci/cloudFPGA/

References

[8] B. Ringlein, F. Abel, A. Ditter, B. Weiss, C. Hagleitner and D. Fey,
"ZRLMPI: A Unified Programming Model for Reconfigurable Heterogeneous Computing Clusters" in 28th International Symposium On Field-Programmable Custom Computing Machines (FCCM), 2020.

[7] B. Ringlein, F. Abel, A. Ditter, B. Weiss, C. Hagleitner and D. Fey,
" **System architecture for network-attached FPGAs in the cloud using partial reconfiguration**," in 29th International Conference on Field Programmable Logic and Applications (FPL), 2019.

[6] F. Abel, J. Weerasinghe, C. Hagleitner, B. Weiss, S. Paredes,
"**An FPGA Platform for Hyperscalers**," in IEEE 25th Annual Symposium on High-Performance Interconnects (HOTI), Santa Clara, CA, pp. 29–32, 2017.

[5] J. Weerasinghe, F. Abel, C. Hagleitner, A. Herkersdorf,
"**Disaggregated FPGAs: Network performance comparison against bare-metal servers, virtual machines and Linux containers,**" in IEEE International Conference on Cloud Computing Technology and Science (CloudCom), Luxembourg, 2016.

[4] J. Weerasinghe, R. Polig, F. Abel,
"**Network-attached FPGAs for data center applications,**" in IEEE International Conference on Field-Programmable Technology (FPT '16), Xian, China, 2016.

[3] J. Weerasinghe, F. Abel, C. Hagleitner, A. Herkersdorf,
"**Enabling FPGAs in hyperscale data centers,**" in IEEE International Conference on Cloud and Big Data Computing (CBDCom), Beijing, China, pp. 1078–1086, 2015.

[2] F. Abel,
"**How do you squeeze 1000 FPGAs into a DC rack?**" online at LinkedIn

[1] The **cloudFPGA project** page at ZRL
https://www.zurich.ibm.com/cci/cloudFPGA/