FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG
FACULTY OF ENGINEERING

Integrating network-attached FPGAs into the cloud using partial reconfiguration

Burkhard Ringlein[1,2], Francois Abel[2], Alexander Ditter[1], Christoph Hagleitner[2], Dietmar Fey[1]

Fourth International Workshop on Heterogeneous High-performance Reconfigurable Computing (H^{2}RC'18)

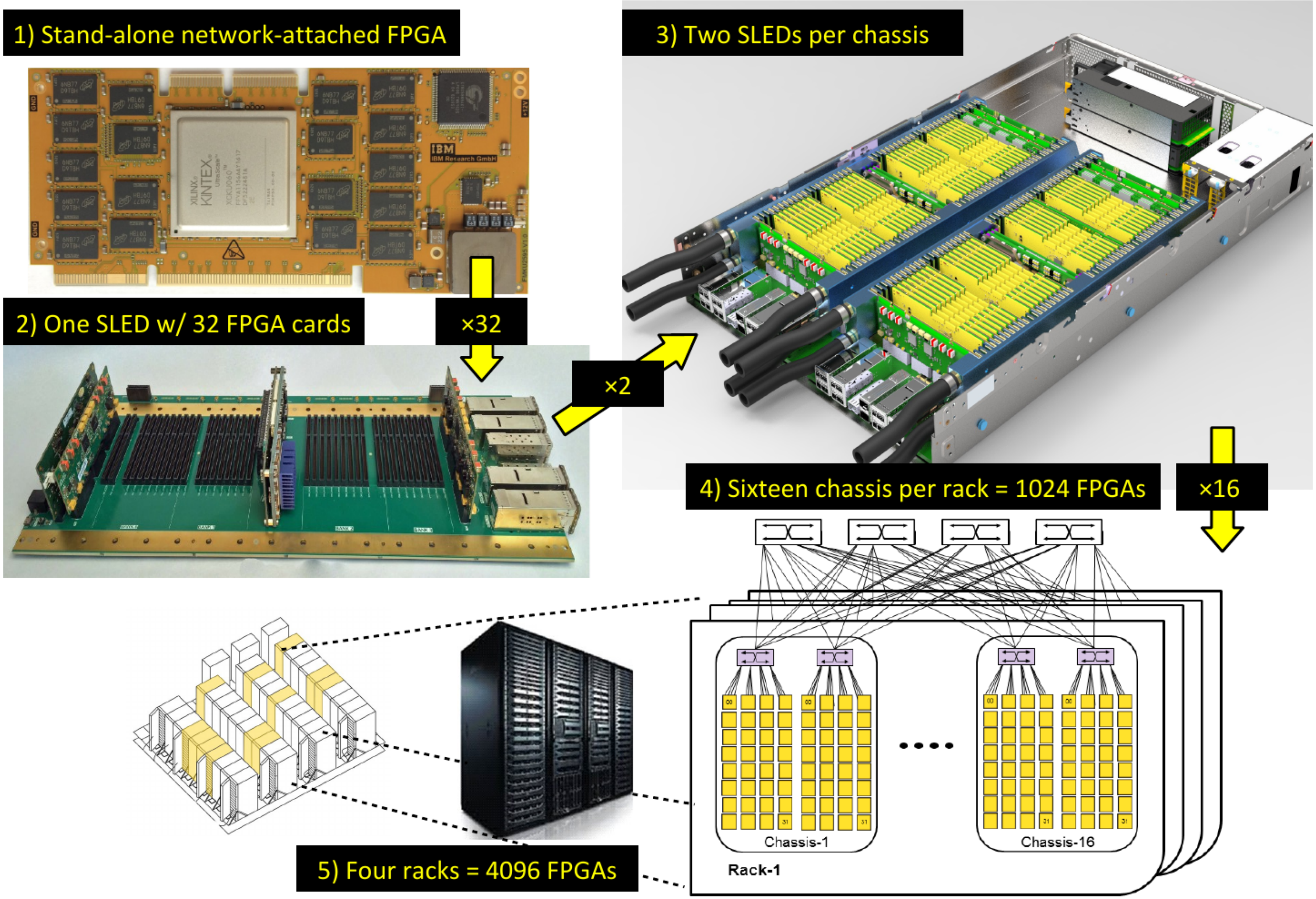
Sunday, November 11, 2018
Dallas, TX

1) *Chair of Computer Architecture, Erlangen, Germany*
2) *IBM Research Zurich, Rüschlikon, Switzerland*

Agenda

1. Background and Motivation
2. Cloud Deployment
3. Management Framework
4. Partial Reconfiguration via Network
5. Conclusion and Outlook

cloudFPGA @ ZRL



1) Stand-alone network-attached FPGA

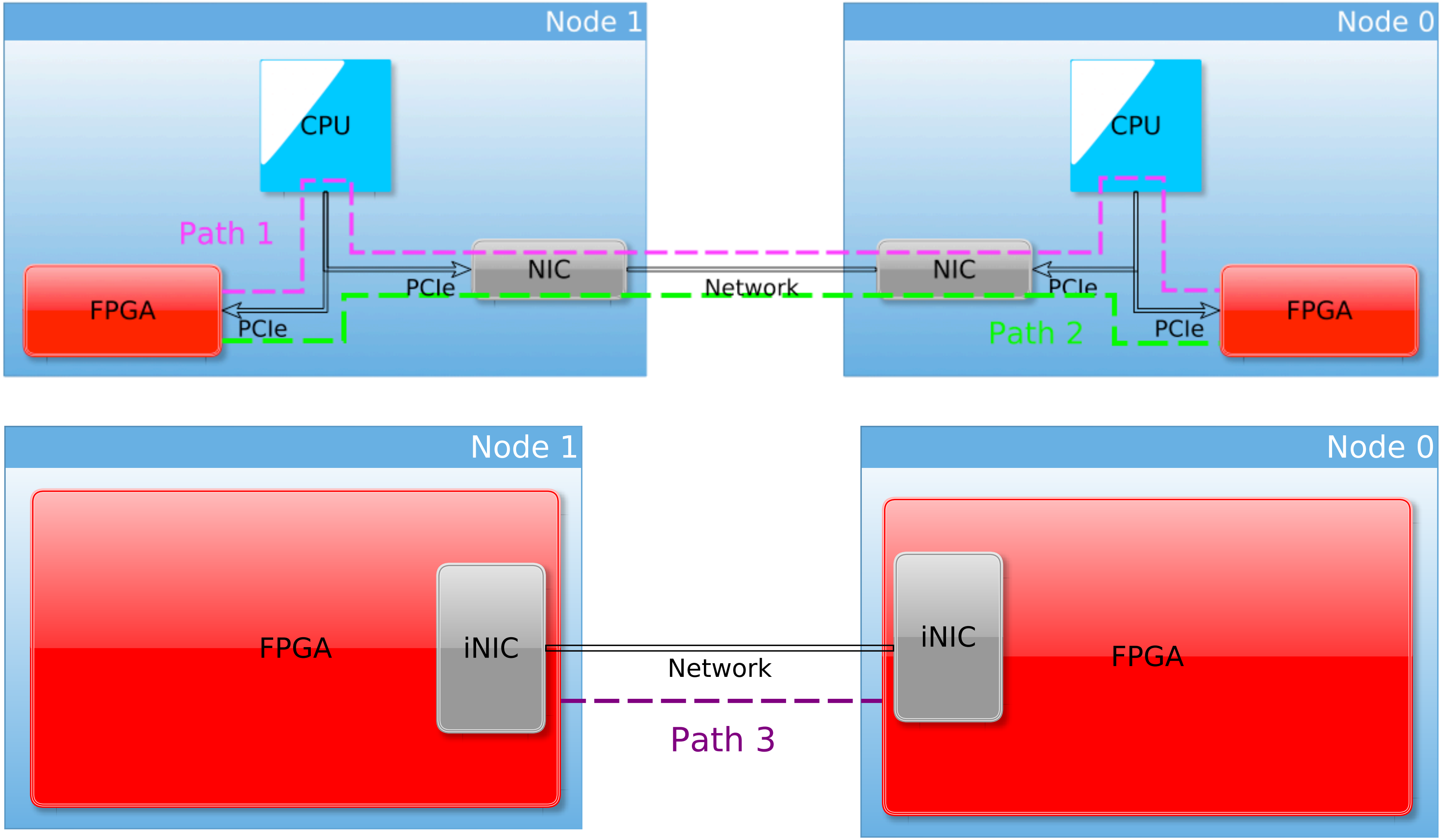
2) One SLED w/ 32 FPGA cards

3) Two SLEDs per chassis

4) Sixteen chassis per rack = 1024 FPGAs

5) Four racks = 4096 FPGAs

Host- vs Network-Attached FPGAs



Cloud Deployment

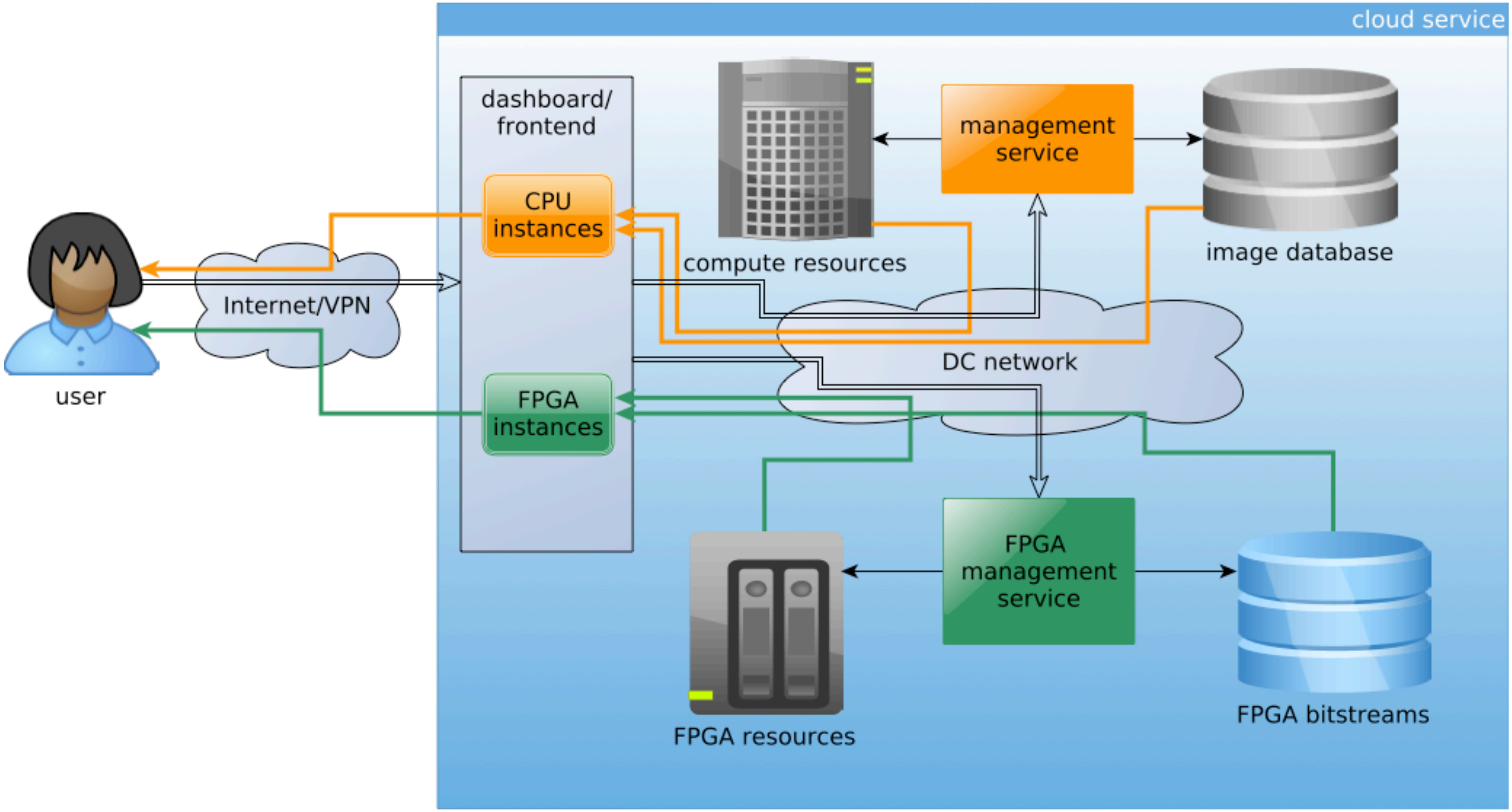
Goals

- Easy-to-use cloud service
- Fast and scalable deployment
- Protection of user specific IP
- Enabling heterogeneous architectures

Constraints

- One communication interface only
- Protection of and separation from DC network
- Standalone FPGA configuration (without any CPU)
- Integration into cloud middleware (OpenStack)

Cloud Deployment Schematic



Protection of Intellectual Property

Transfer of user code onto FPGAs

Keeping algorithms and business secrets protected as users

- may not wish to expose algorithmic details of their codes,
- may not be allowed to upload source code due to corporate policies,
- might want to be able to deploy pre-compiled bitstreams.

The “off-site compilation” of an FPGA design for deployment would take hours – every time!

→ This would *neither* be *fast nor scalable*

➔ Architecture must be able to deploy and run applications based on their bitstreams

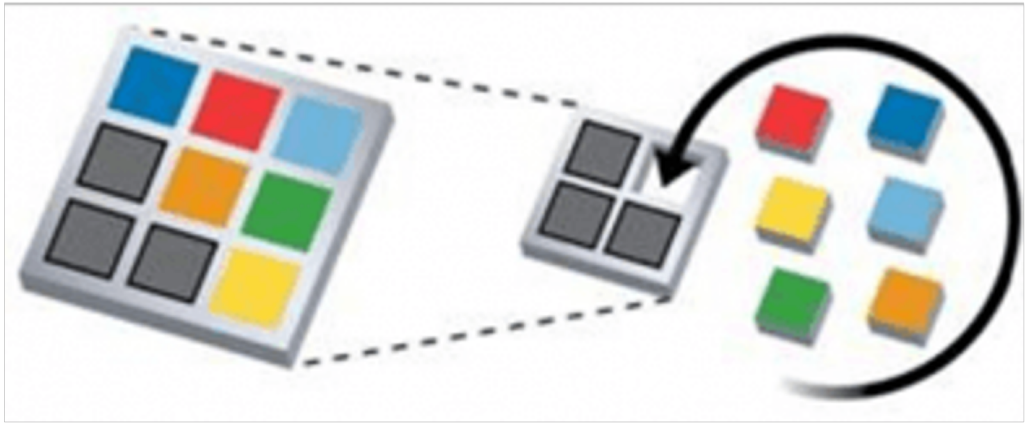
➔ Applications must be independent of the final FPGA / know the target at compile time

Privilege Separation within the FPGA

Q: How to deploy user synthesized designs?

A: Partial Reconfiguration (PR)

- allows a design to keep its internal states, while parts are updated
- makes sure that only the user logic can be modified
- creates a clear separation between mgnt. / app. logic and network
- enables the potential for multitenancy and "virtualization"
- leads to overall faster reconfiguration



Management of Disaggregated FPGAs

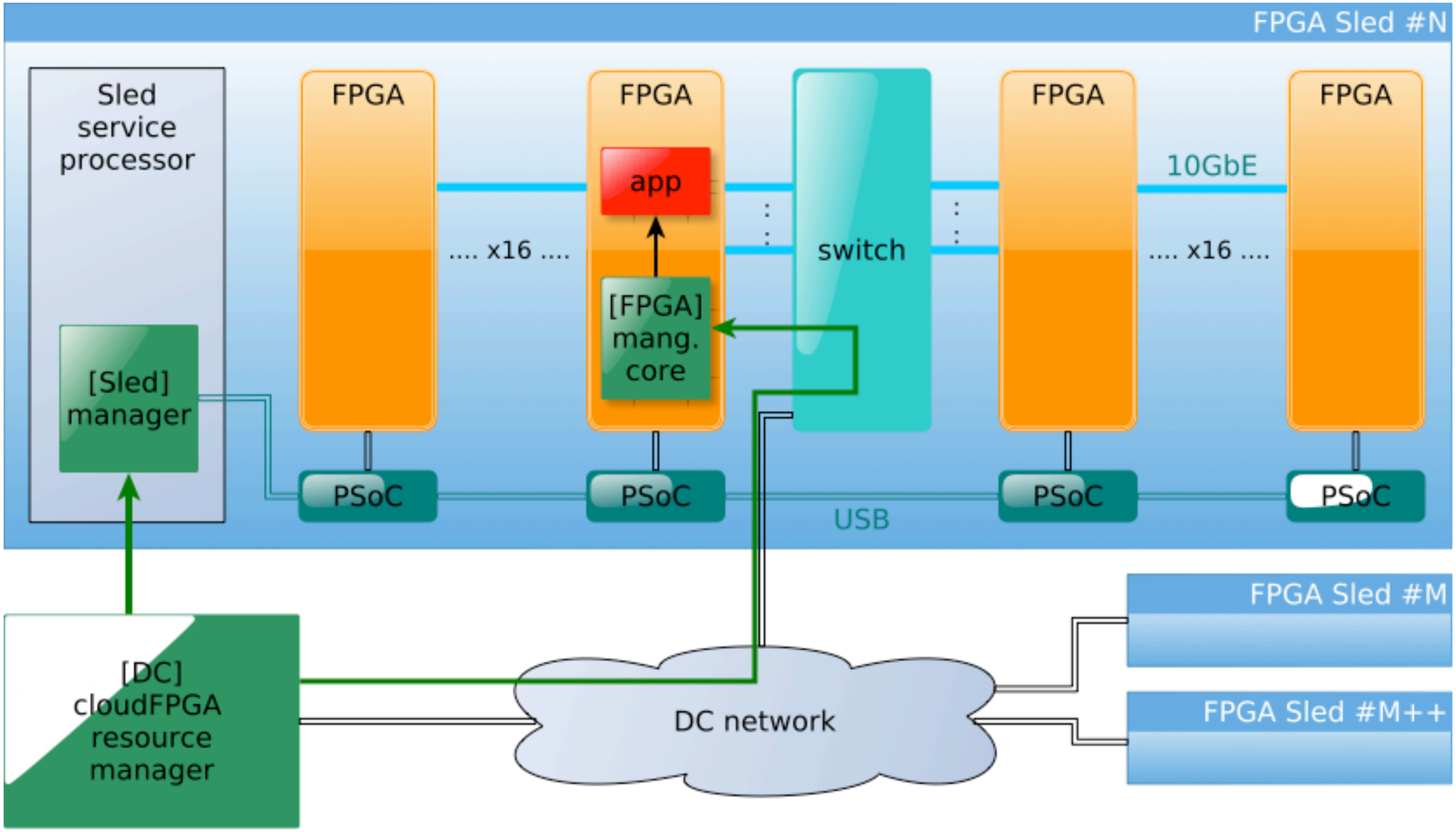
Tasks for standalone network-attached FPGAs

- Acquisition of the necessary number of FPGAs (of requested type)
- Powering FPGAs on/off (before and after usage)
- Deployment of application binary on all devices
- Set-up network for user specific application

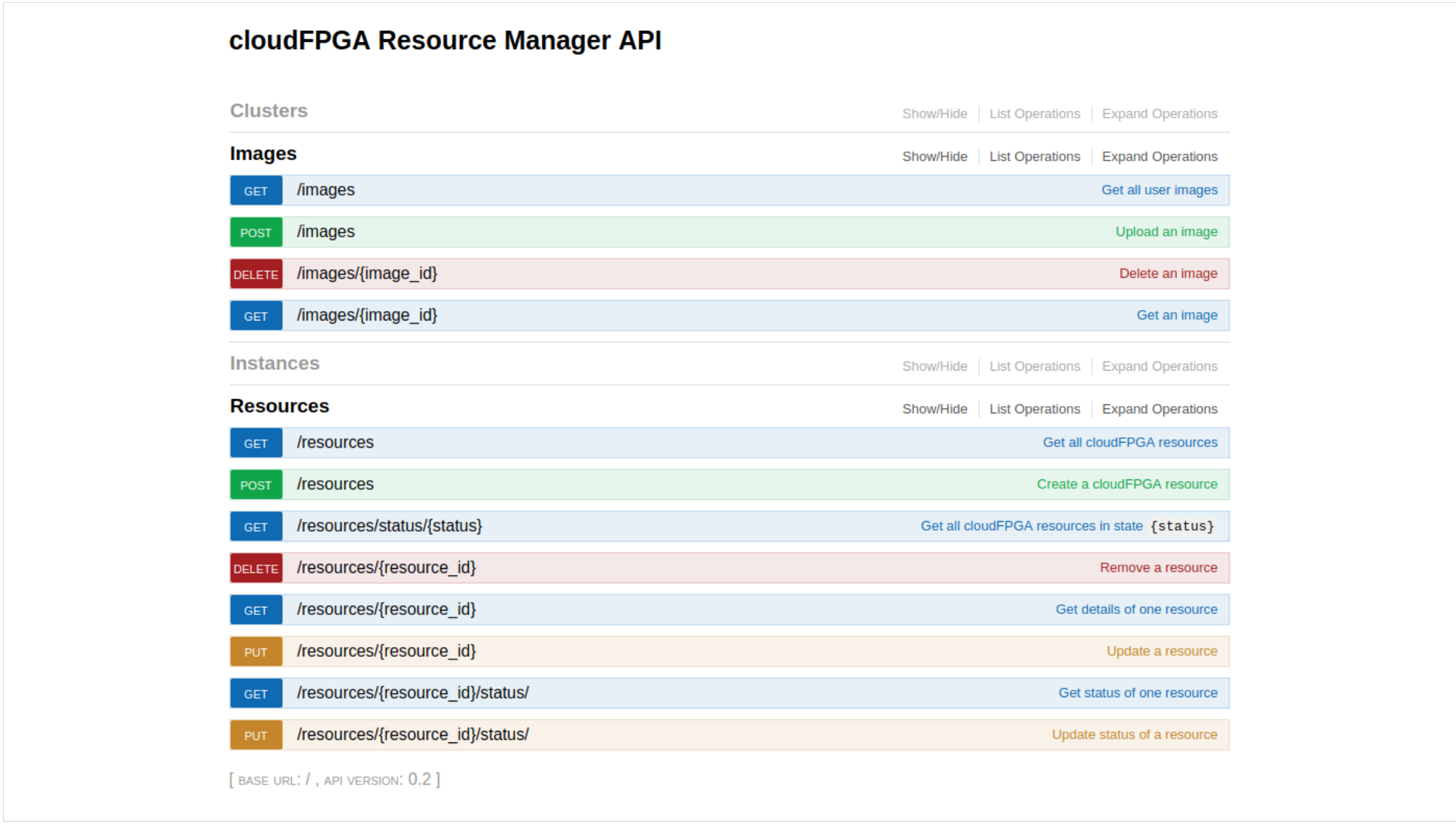
Consider different levels for above tasks

DC level	acquire (& release) FPGAs and IP-addresses
	store bitstreams
Rack level	power FPGAs on/off
	program IP-addresses
FPGA level	configure application bitstream
	set up network routing and provide node id

Management Framework



Communication between hardware and software via RESTful API



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

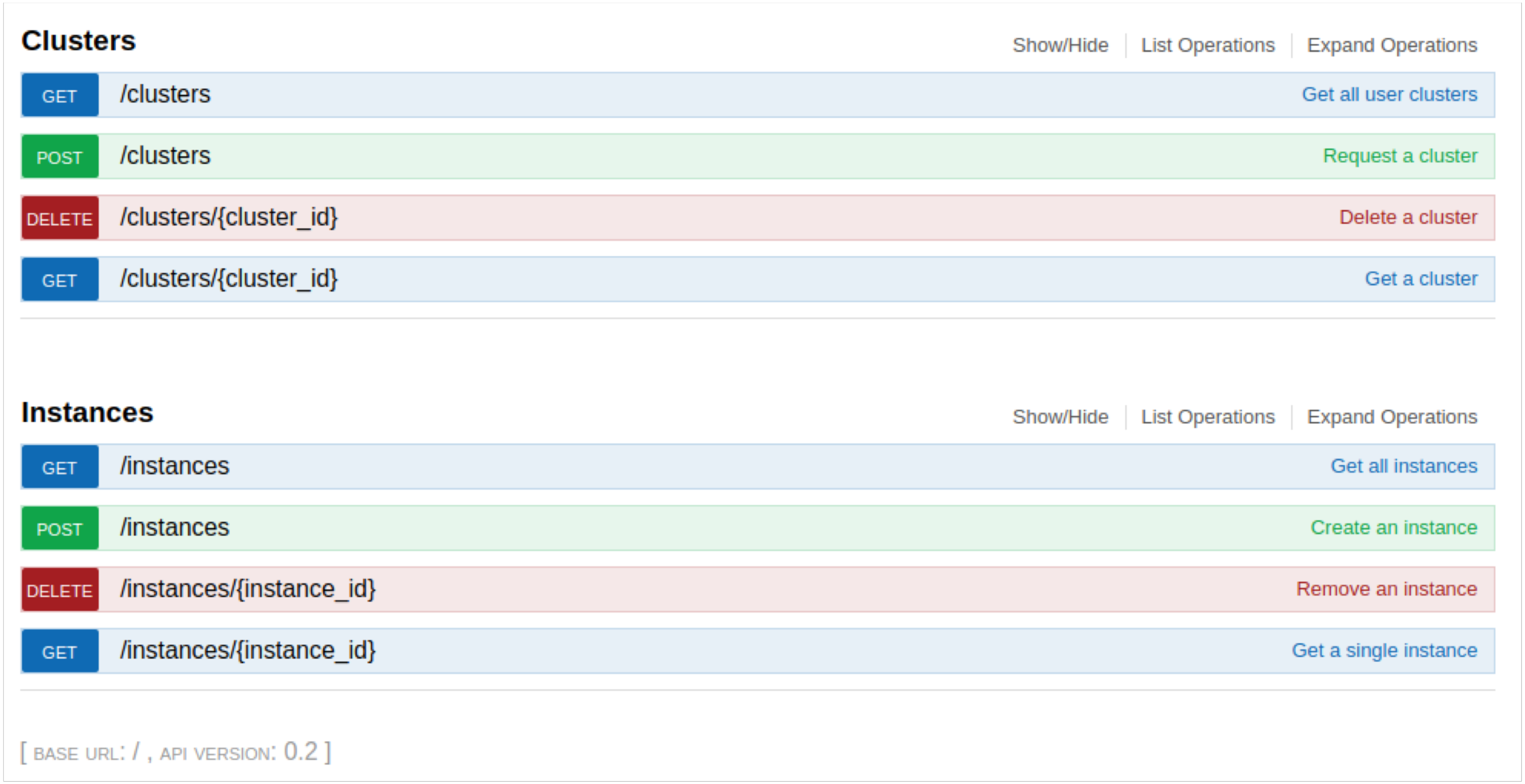
Show/Hide | List Operations | Expand Operations

Resources

Show/Hide | List Operations | Expand Operations

GET	/resources	Get all cloudFPGA resources
POST	/resources	Create a cloudFPGA resource
GET	/resources/status/{status}	Get all cloudFPGA resources in state `{status}`
DELETE	/resources/{resource_id}	Remove a resource
GET	/resources/{resource_id}	Get details of one resource
PUT	/resources/{resource_id}	Update a resource
GET	/resources/{resource_id}/status/	Get status of one resource
PUT	/resources/{resource_id}/status/	Update status of a resource

[BASE URL: / , API VERSION: 0.2]



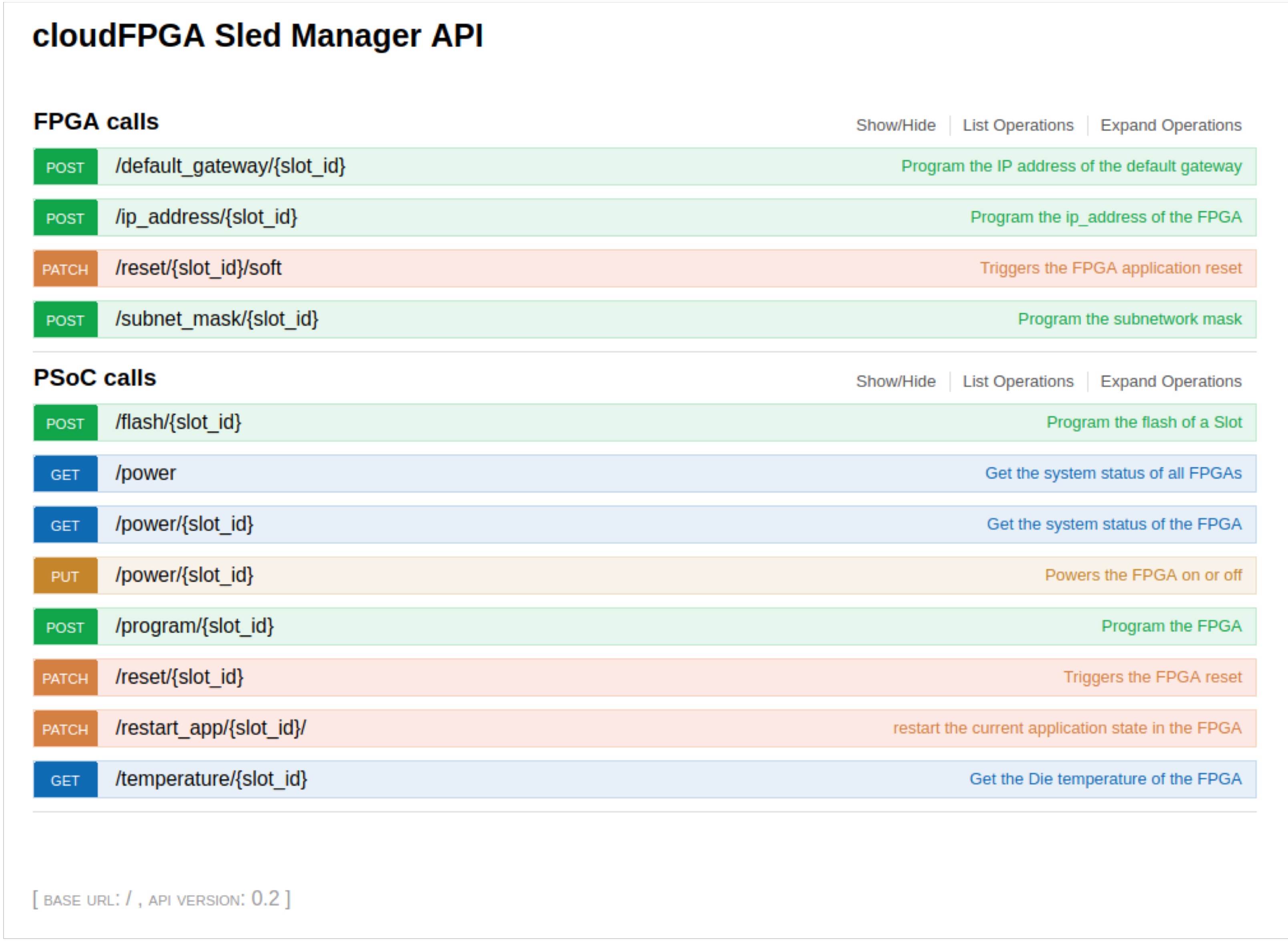
Clusters

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

Instances

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

[BASE URL: / , API VERSION: 0.2]



cloudFPGA Sled Manager API

FPGA calls

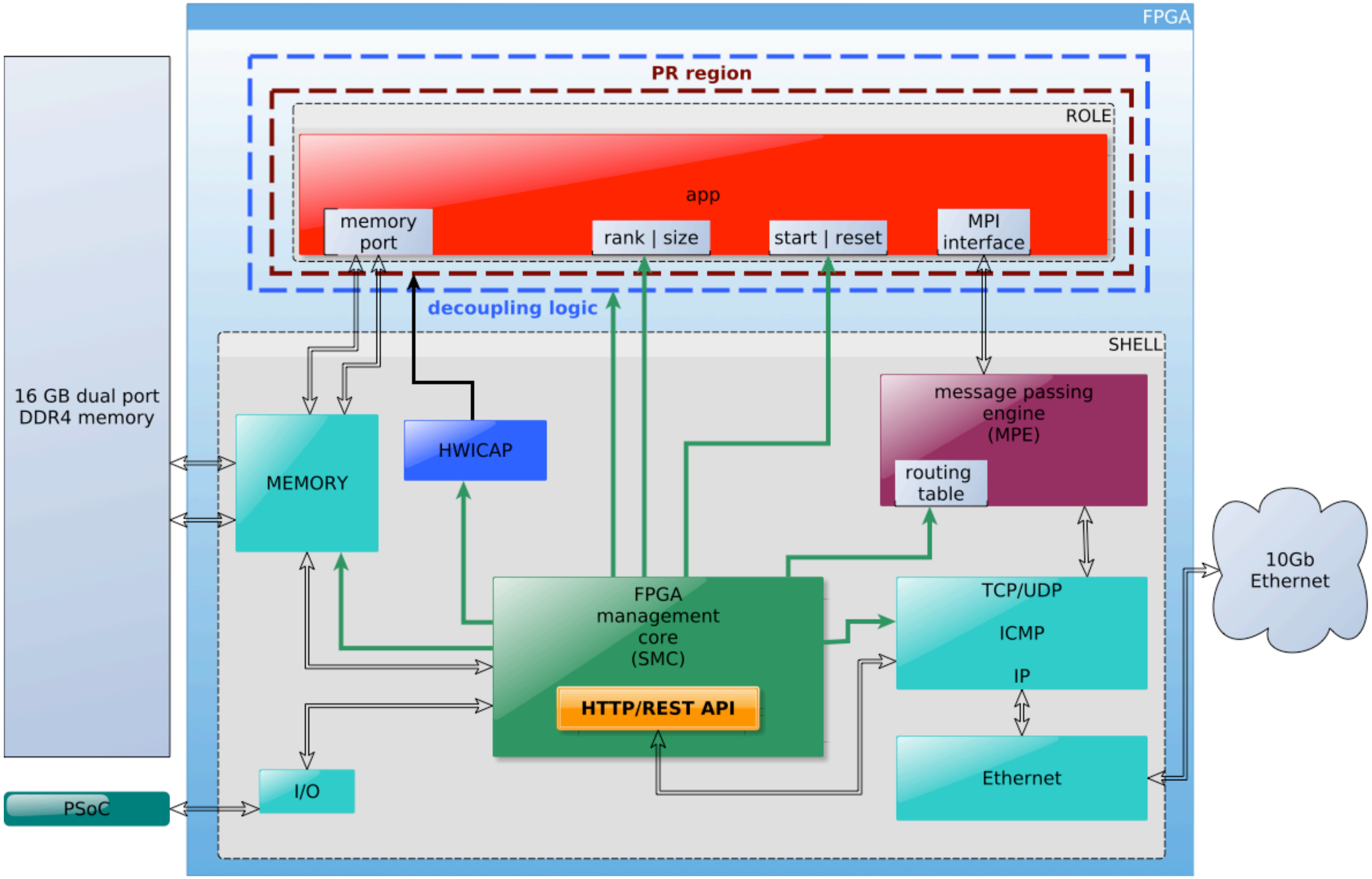
POST	/default_gateway/{slot_id}	Program the IP address of the default gateway
POST	/ip_address/{slot_id}	Program the ip_address of the FPGA
PATCH	/reset/{slot_id}/soft	Triggers the FPGA application reset
POST	/subnet_mask/{slot_id}	Program the subnetwork mask

PSoC calls

POST	/flash/{slot_id}	Program the flash of a Slot
GET	/power	Get the system status of all FPGAs
GET	/power/{slot_id}	Get the system status of the FPGA
PUT	/power/{slot_id}	Powers the FPGA on or off
POST	/program/{slot_id}	Program the FPGA
PATCH	/reset/{slot_id}	Triggers the FPGA reset
PATCH	/restart_app/{slot_id}/	restart the current application state in the FPGA
GET	/temperature/{slot_id}	Get the Die temperature of the FPGA

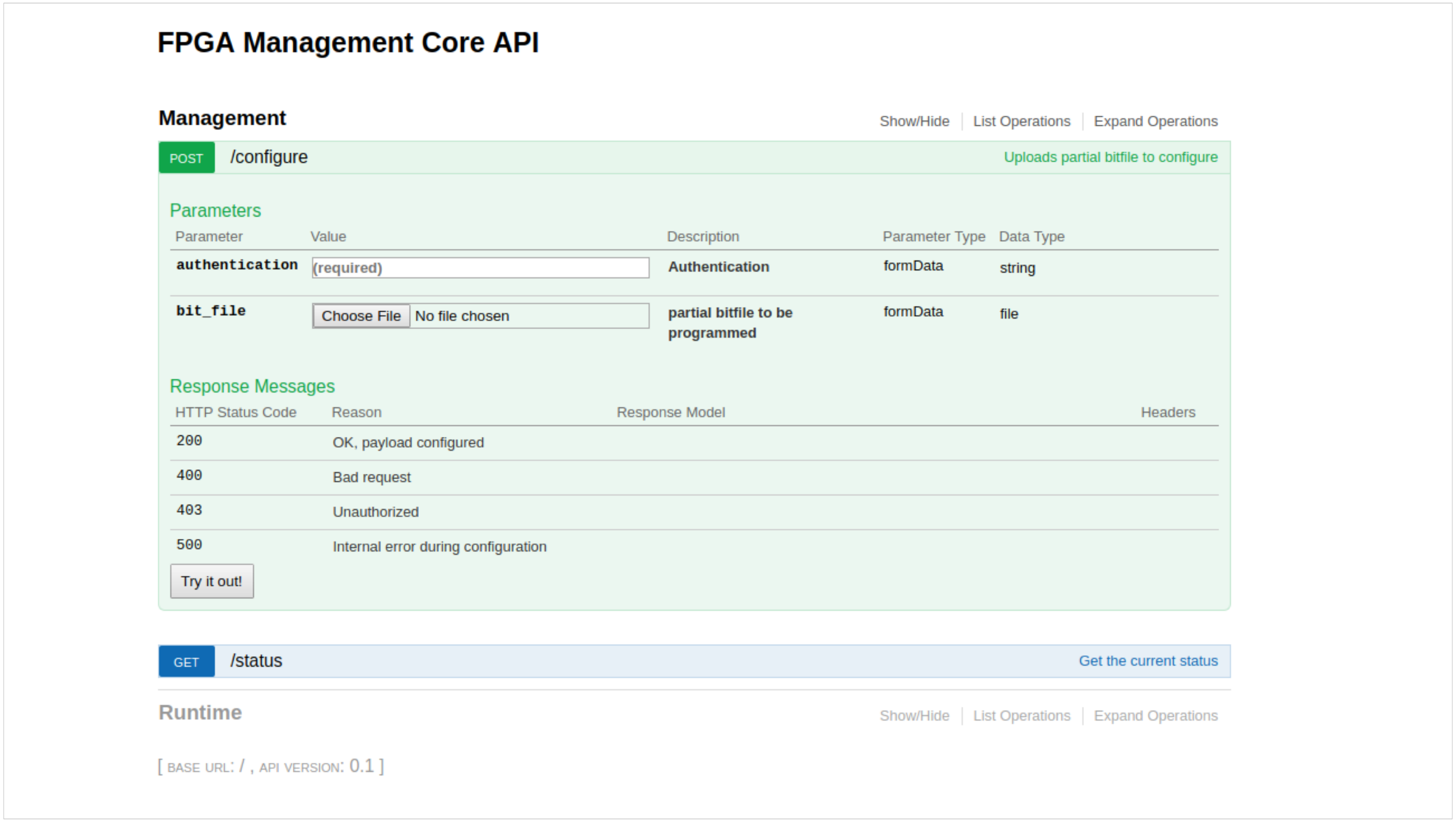
[BASE URL: / , API VERSION: 0.2]

Partial Reconfiguration via Network



➔ RESTful management core inside FPGA for distributed management

FPGA Management Core API (1/2)



FPGA Management Core API

Management

POST /configure — Uploads partial bitfile to configure

Parameters

Parameter	Value	Description	Parameter Type	Data Type
authentication	(required)	Authentication	formData	string
bit_file	Choose File No file chosen	partial bitfile to be programmed	formData	file

Response Messages

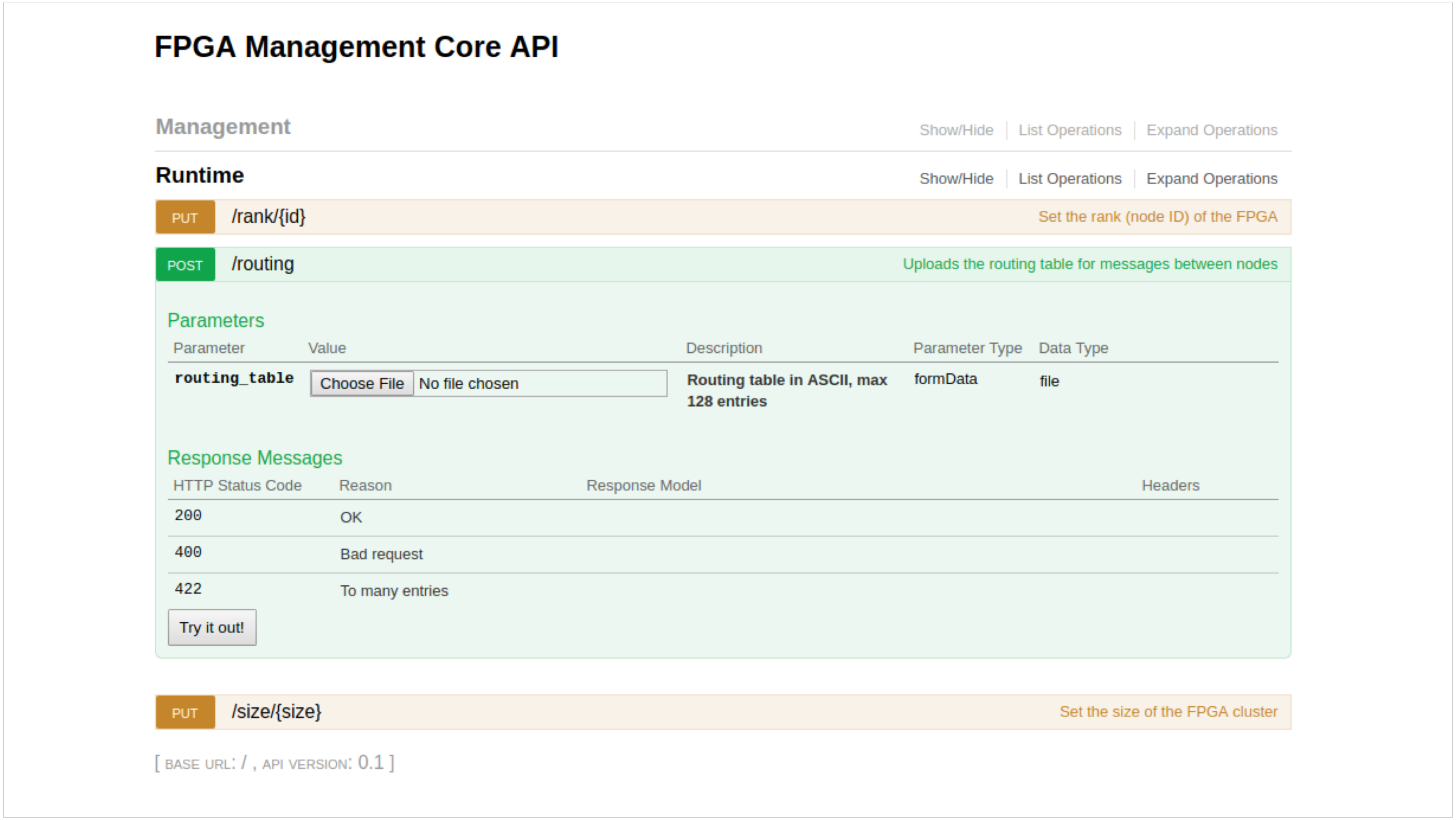
HTTP Status Code	Reason	Response Model	Headers
200	OK, payload configured		
400	Bad request		
403	Unauthorized		
500	Internal error during configuration		

GET /status — Get the current status

Runtime

[BASE URL: / , API VERSION: 0.1]

FPGA Management Core API (2/2)



FPGA Management Core API

Management — Show/Hide | List Operations | Expand Operations

Runtime — Show/Hide | List Operations | Expand Operations

PUT /rank/{id} — Set the rank (node ID) of the FPGA

POST /routing — Uploads the routing table for messages between nodes

Parameters

Parameter	Value	Description	Parameter Type	Data Type
`routing_table`	Choose File No file chosen	**Routing table in ASCII, max 128 entries**	formData	file

Response Messages

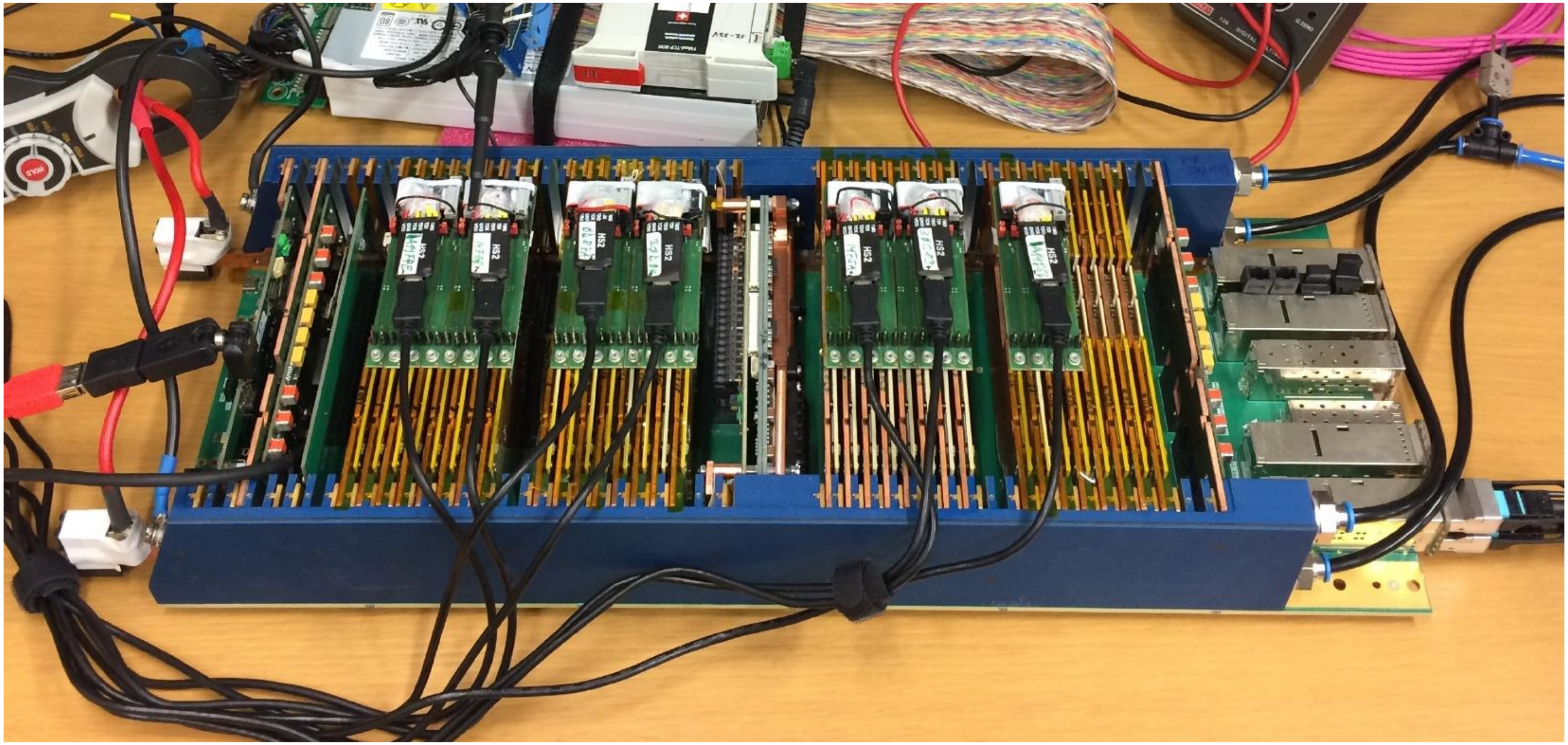
HTTP Status Code	Reason	Response Model	Headers
200	OK		
400	Bad request		
422	To many entries		

Try it out!

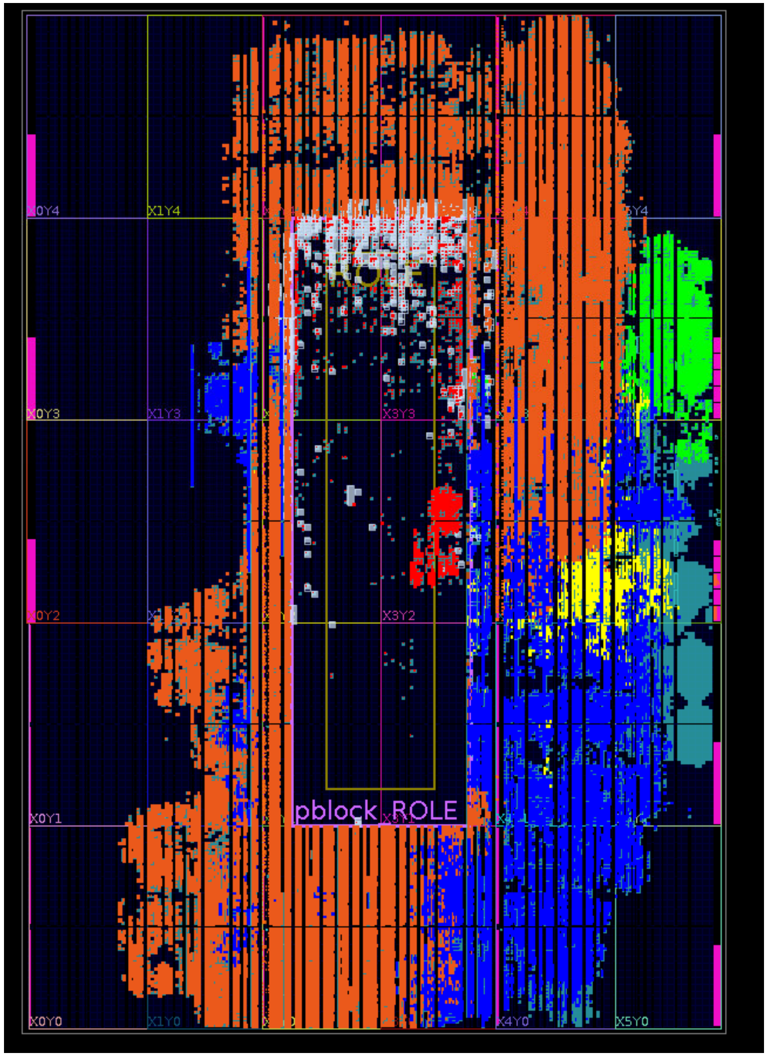
PUT /size/{size} — Set the size of the FPGA cluster

[BASE URL: / , API VERSION: 0.1]

Proof of Concept – Prototype in the Lab



Proof of Concept and Implementation



SMC, HWICAP and Decoupler
MPE
UDP, TCP, IP and ICMP
DDR4 Memory controler (both)
Role

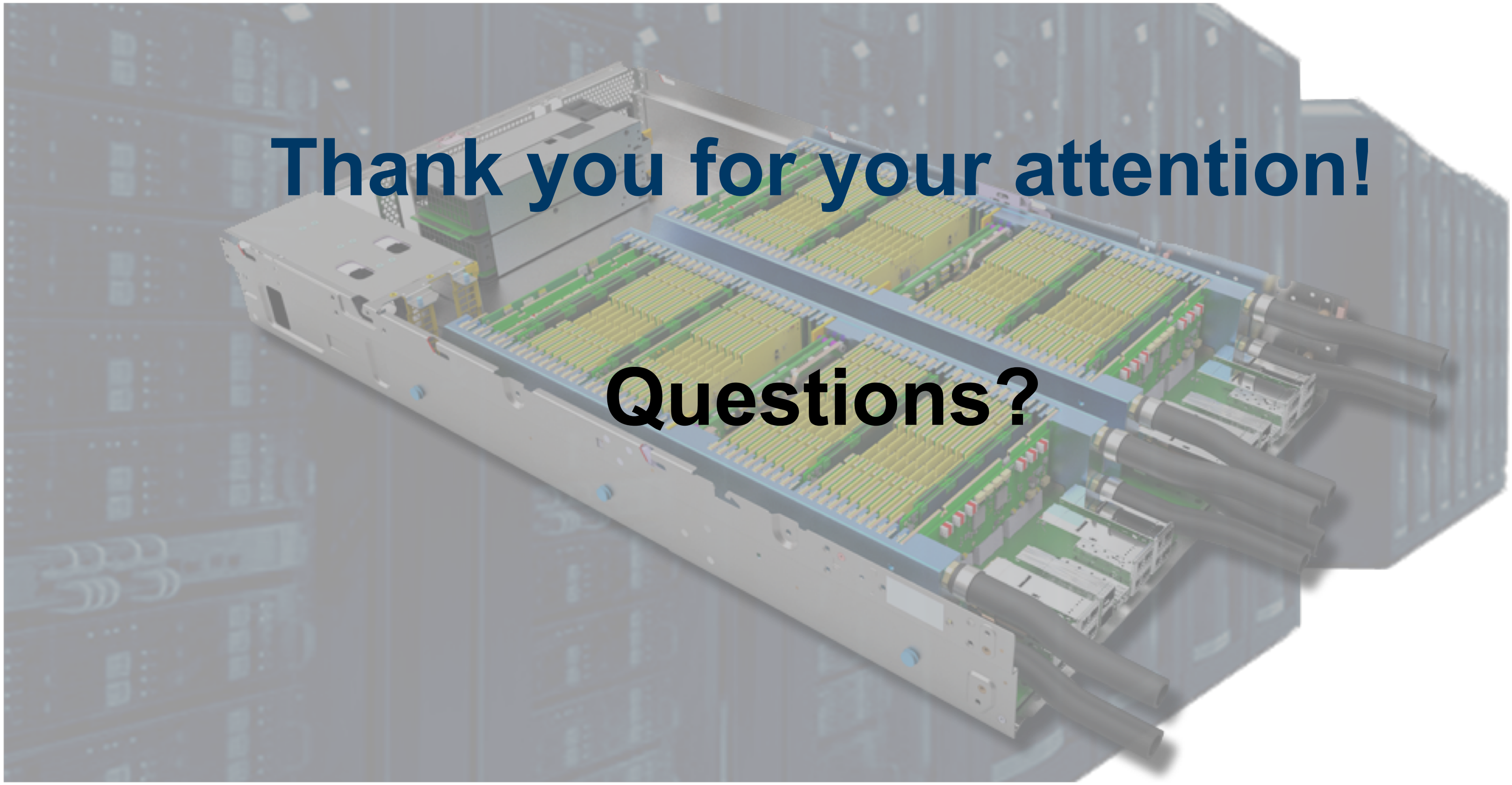
Conclusion and outlook

Proposition of an architecture to

- acquire network-attached FPGAs,
- execute distributed applications,
- protect user specific IP and
- support large scale deployment in DCs.

Future work

- Fine tune and extend management service
- Extend MPI implementation for FPGAs
- Evaluate FPGAs for "Function-as-a-Service" computation



Thank you for your attention!

Questions?

Contact:

Alexander Ditter
alexander.ditter@fau.de

Burkhard Ringlein
NGL@zurich.ibm.com