Protocol Buffer DPU Offloading in the RPC Datapath

SC24 Workshop: Communication, I/O, and Storage at Scale on Next-Generation Platforms - Scalable Infrastructures

Raphaël Frantz, Jerónimo Sánchez García, Marcin Copik, Idelfonso Tafur Monroy, Juan José Vegas Olmos, Gil Bloch, Salvatore Di Girolamo

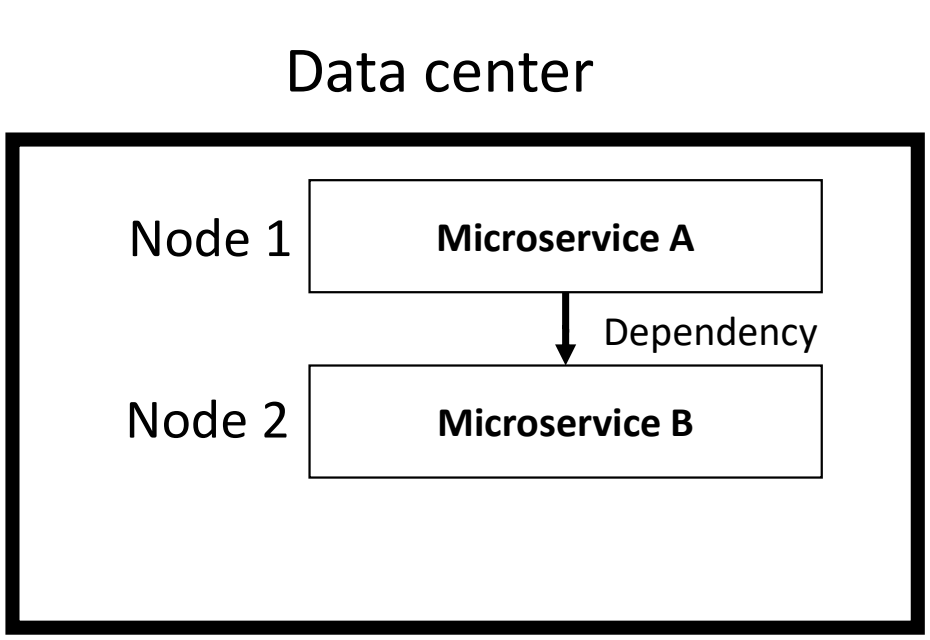
Quantum Terahertz Systems, Electro-Optical Communications, Department of Electrical Engineering

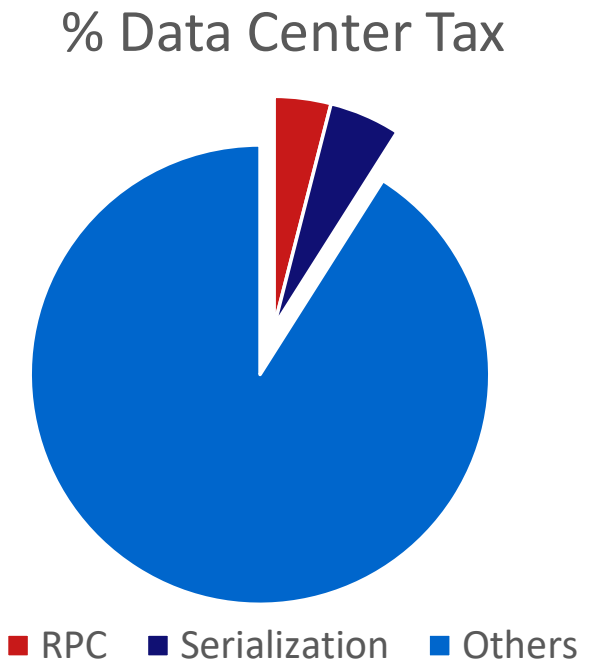
Outline

1. Background
2. Designs
3. Benchmarks
4. Conclusion

Microservice Architecture

- Today's applications are **split** in independent "microservices"
- Communication is done with RPCs





Kanev, S. et al., 2016
Profiling a warehouse-scale computer

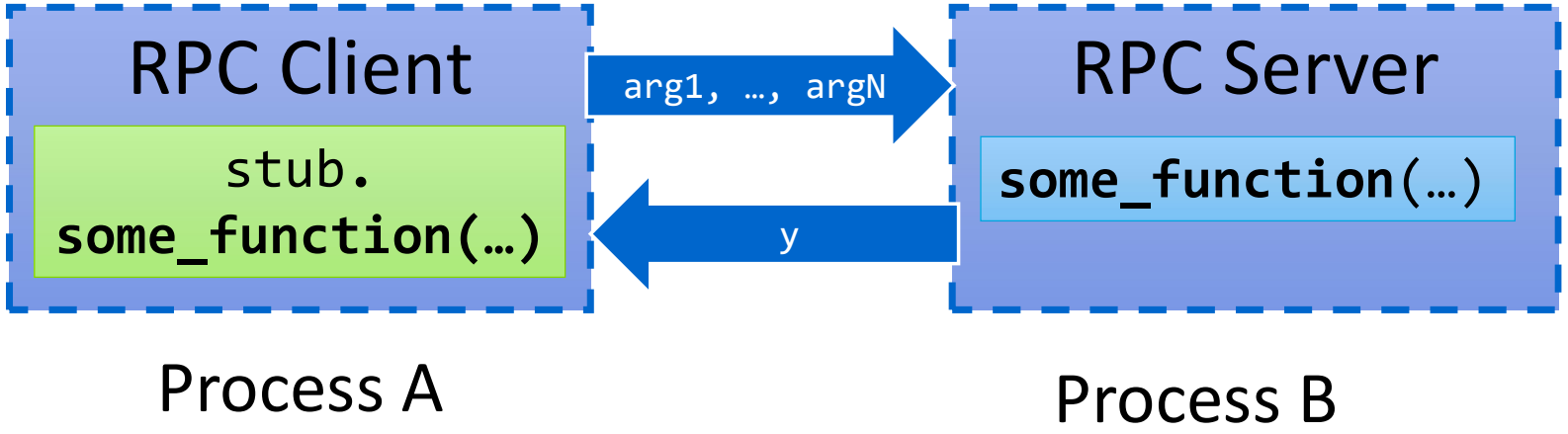
Remote Procedure Calls (RPCs)

High-level abstraction of communication with function calls

In code:

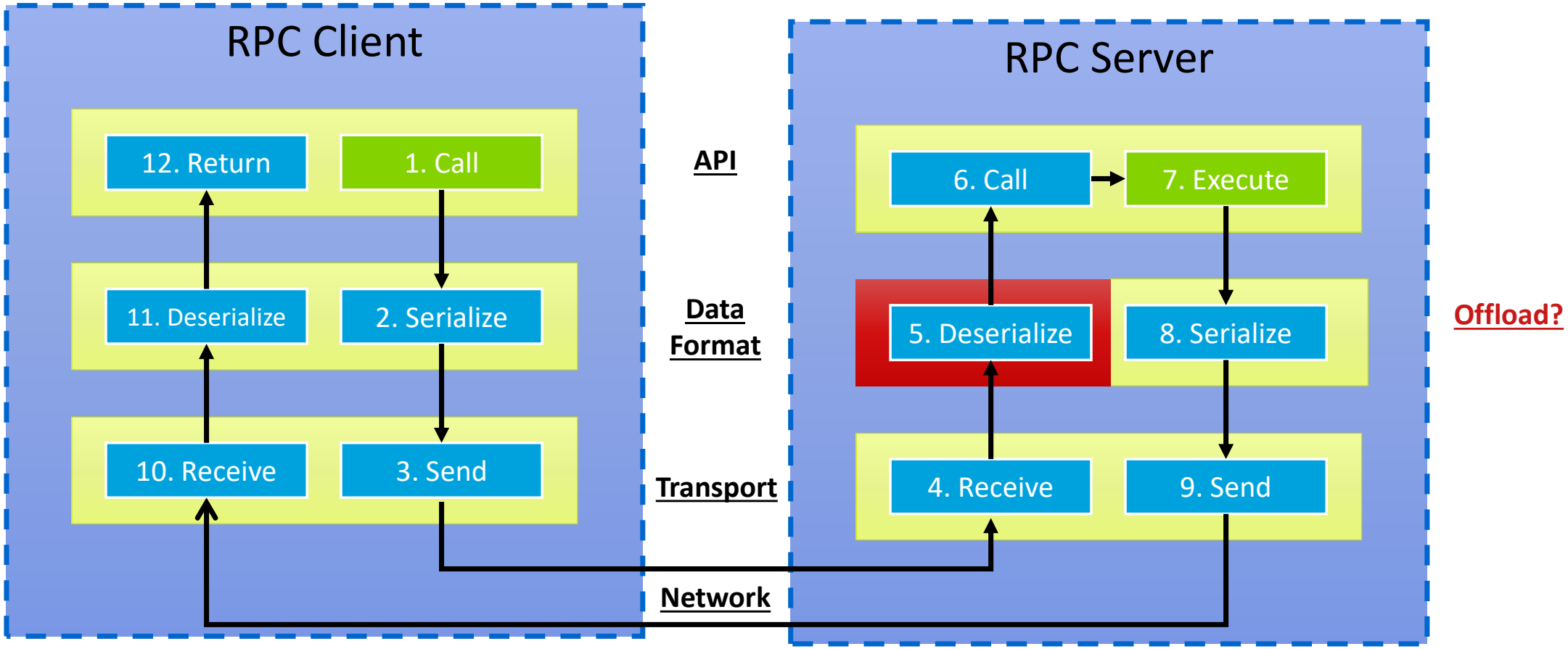
```
y = stub.some_function(arg1, …, argN)
```

At runtime:



RPC Datapath

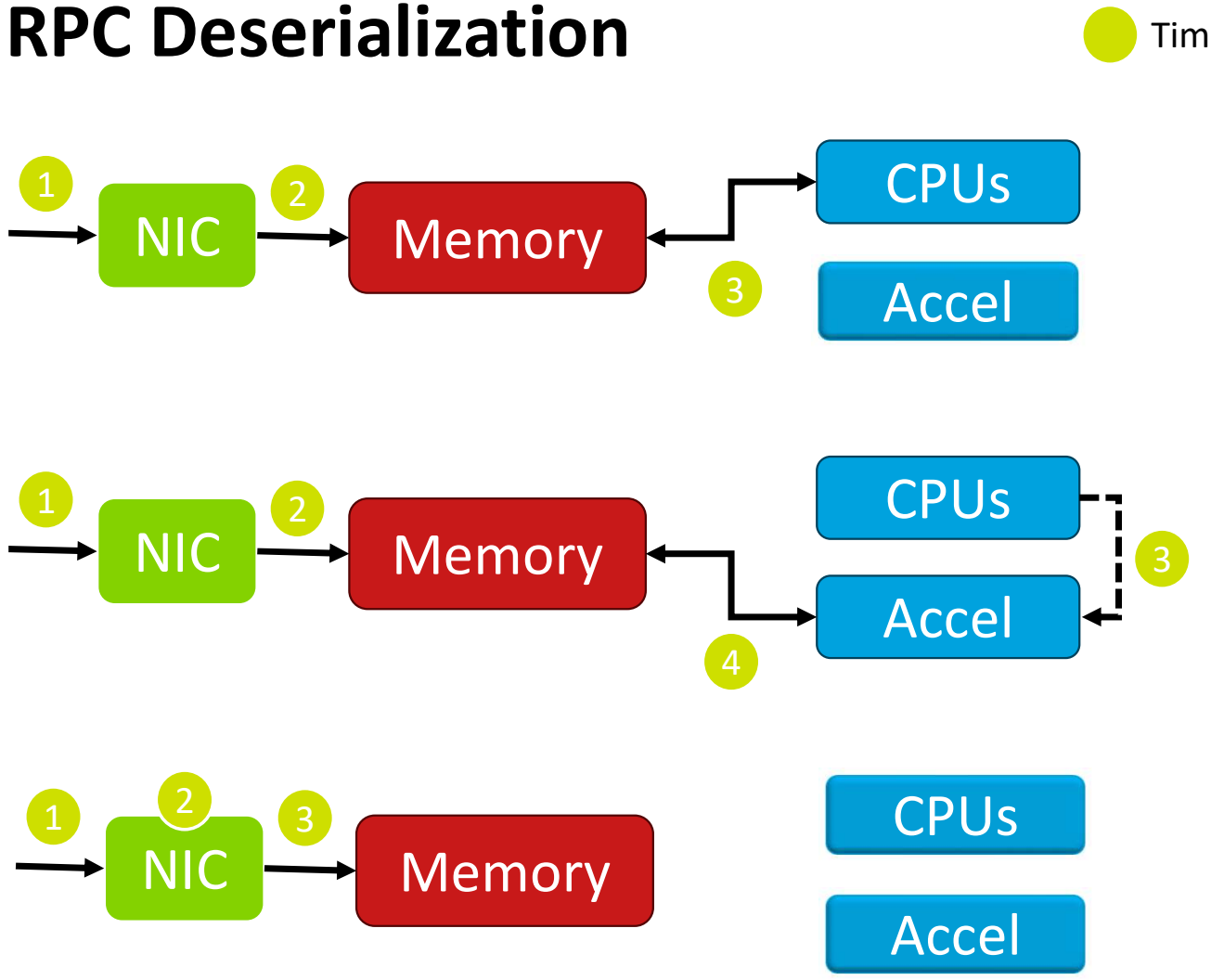
Exposed to the User | RPC Framework



QUARC

TU/e

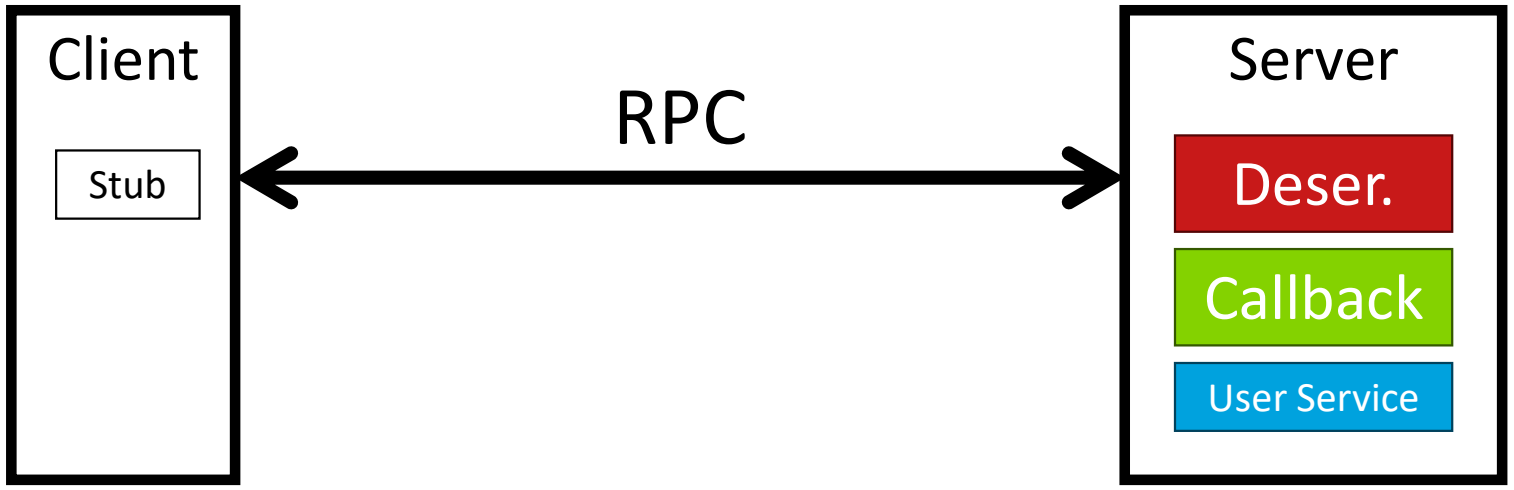
Strategies to Offload RPC Deserialization



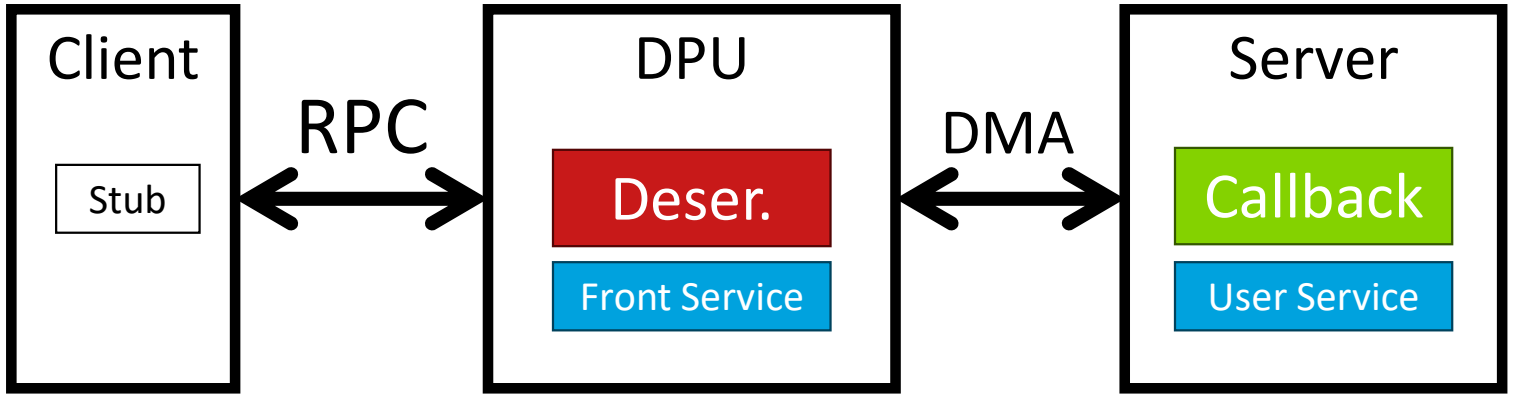
Cao, S. et al., 2022.
Accelerating Data Serialization/Deserialization Protocols with In-Network Compute

Our Solution to Offload RPC Deserialization

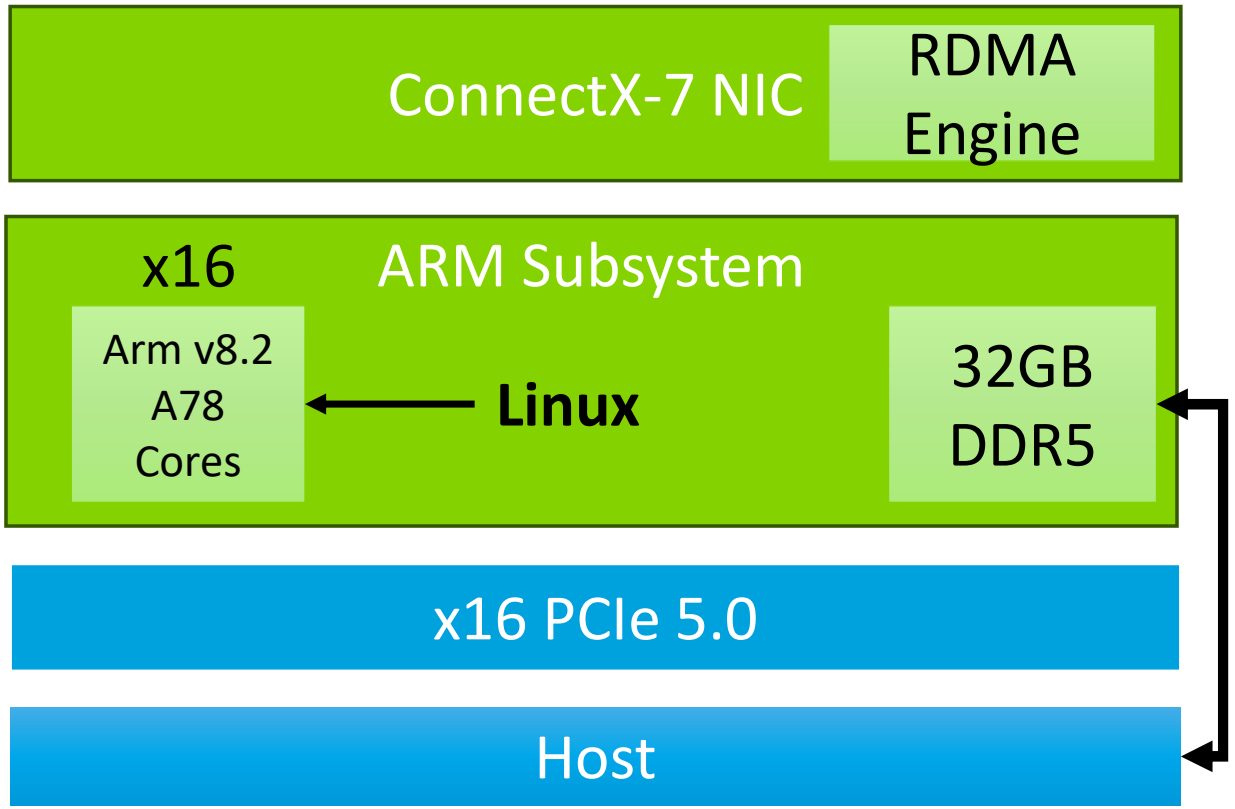
No offloading



Offloading

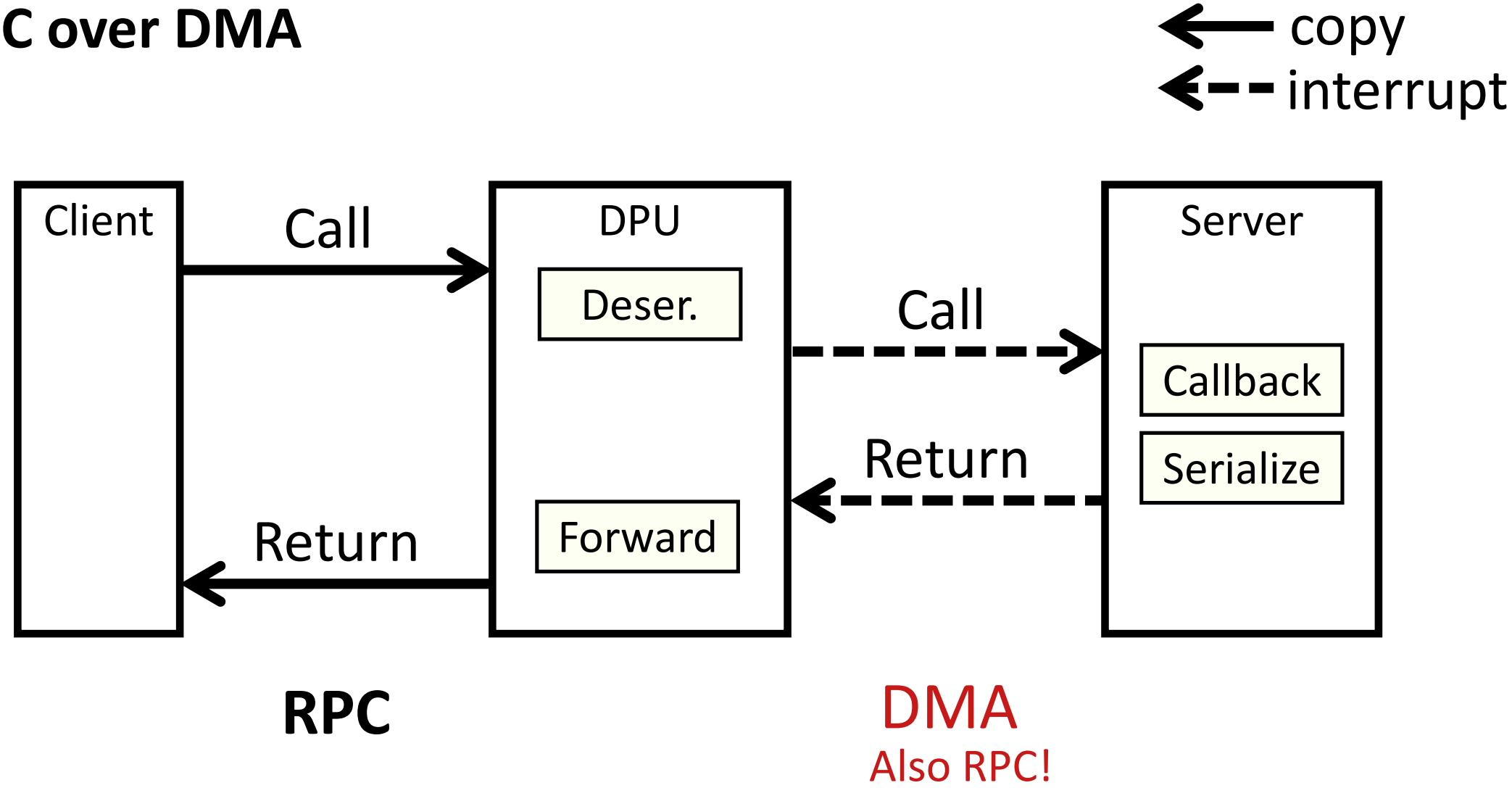


BlueField-3 Data Processing Units (DPUs)

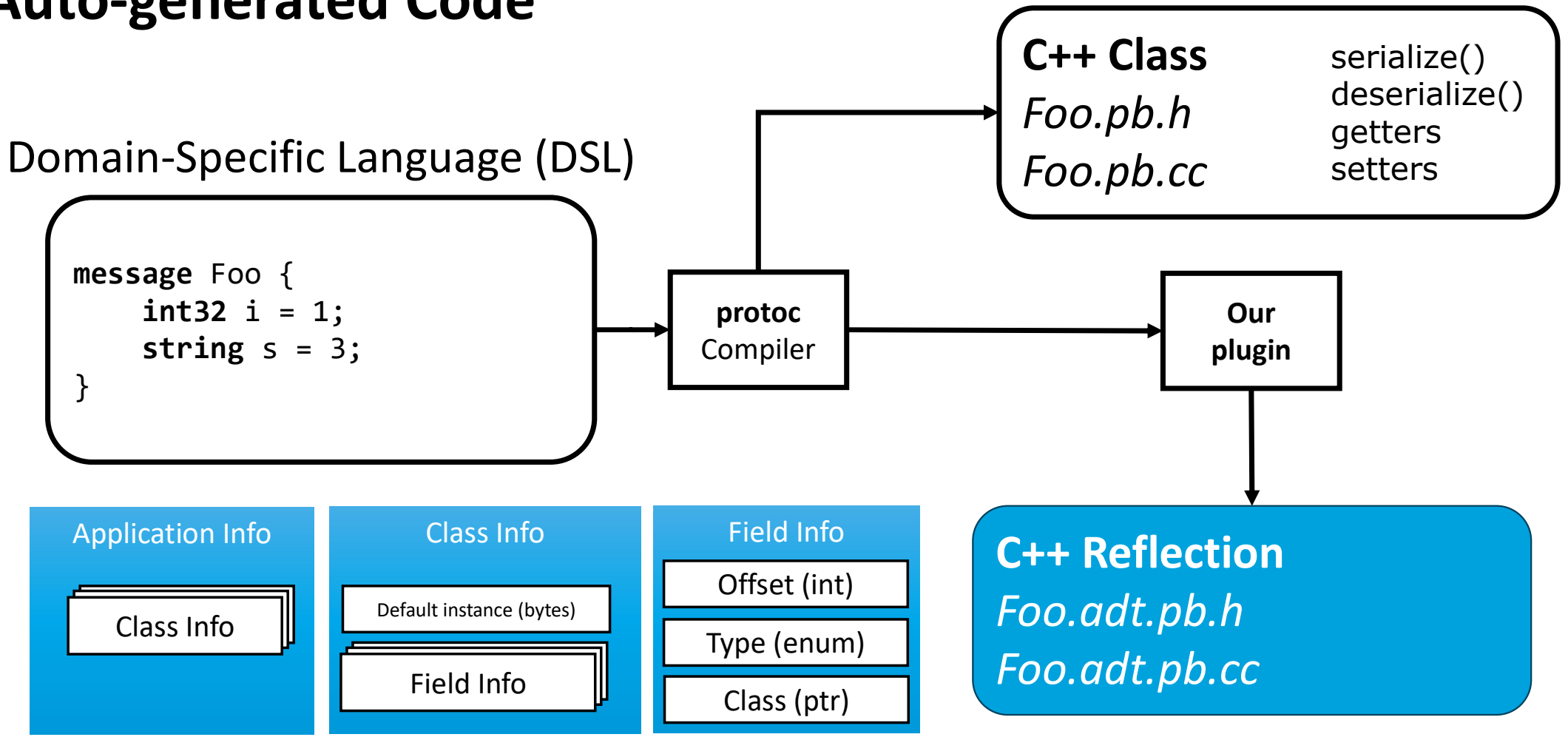


Direct Memory Access

RPC over DMA

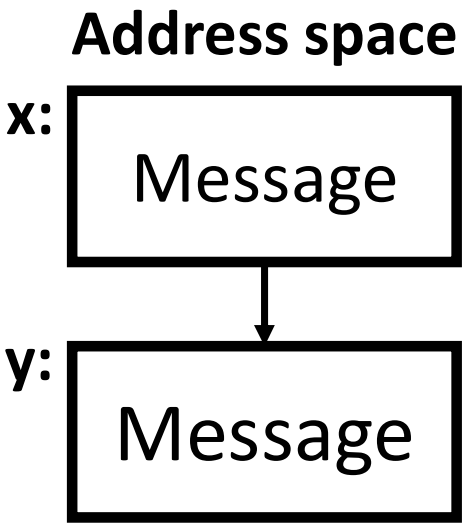

copy
interrupt
Client
Call
DPU
Deser.
Call
Server
Callback
Serialize
Return
Forward
Return
RPC
DMA
Also RPC!

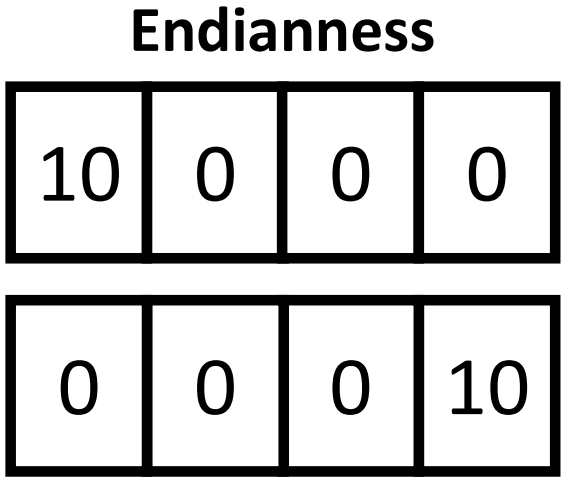
Auto-generated Code

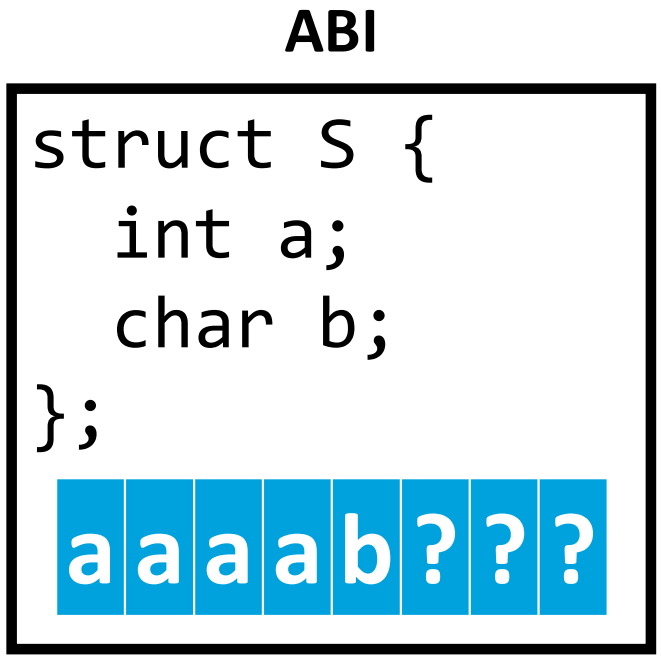


Challenges with offloading Deserialization

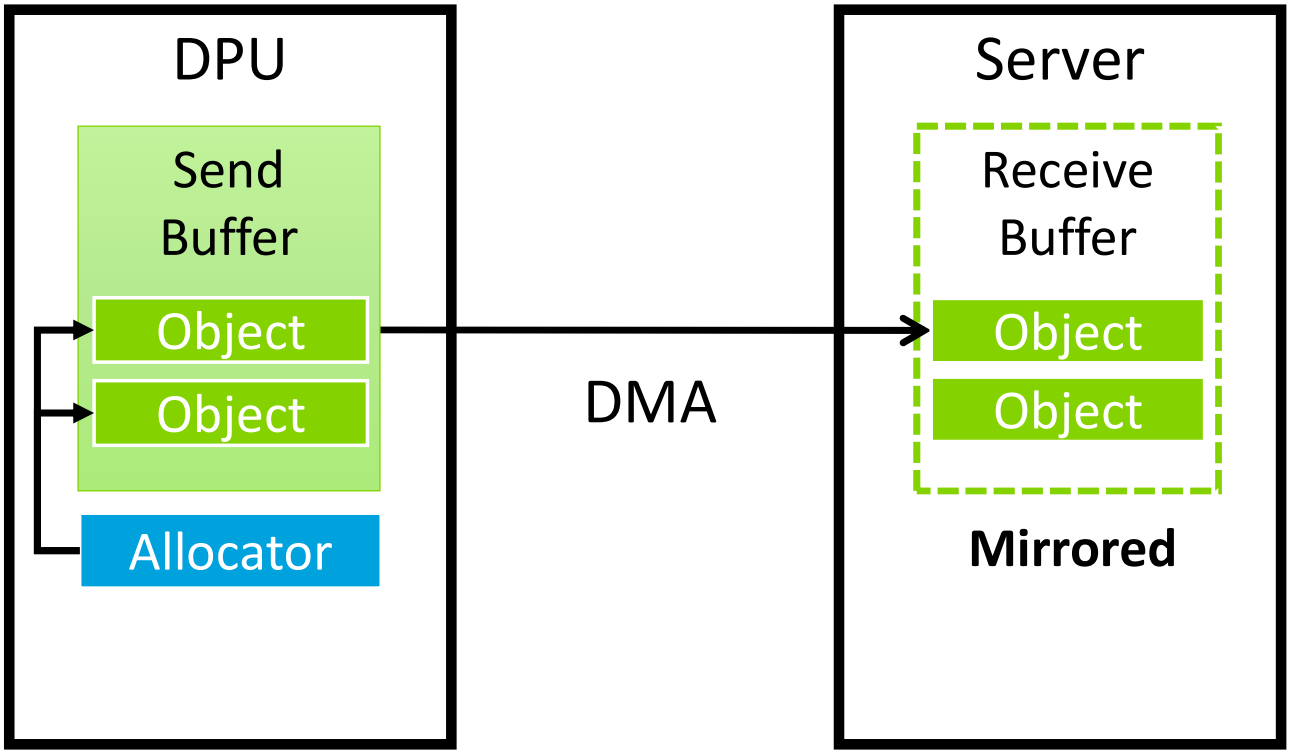
1. Address space ☹
2. Application Binary Interface (ABI)
   - DPU ABI = Server ABI = Itanium ☺

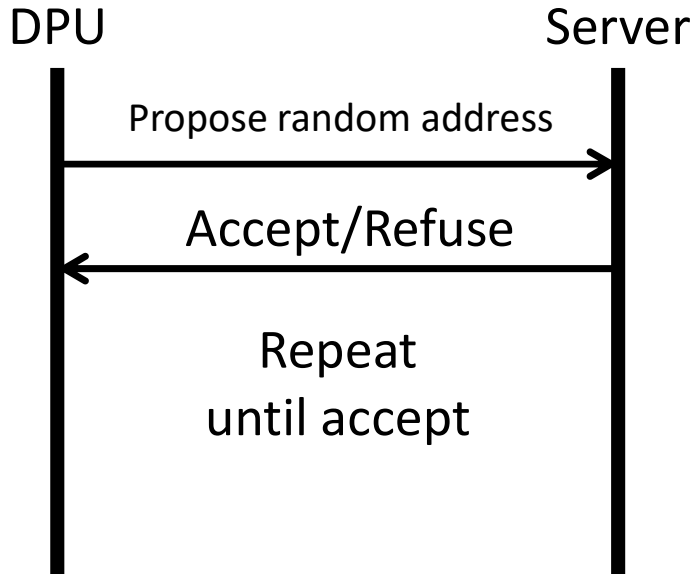






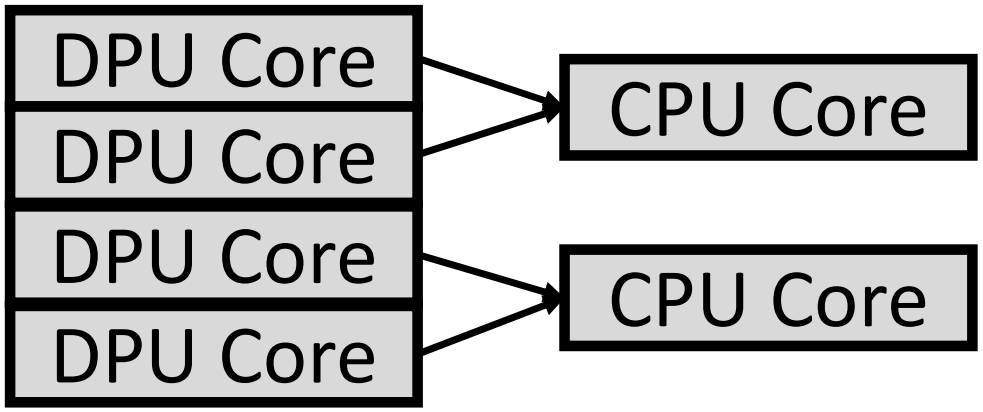
Shared Address Space



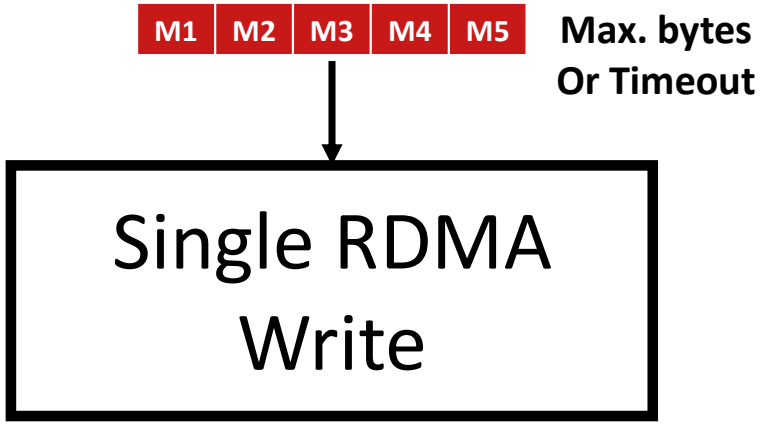


DMA-based RPC Protocol

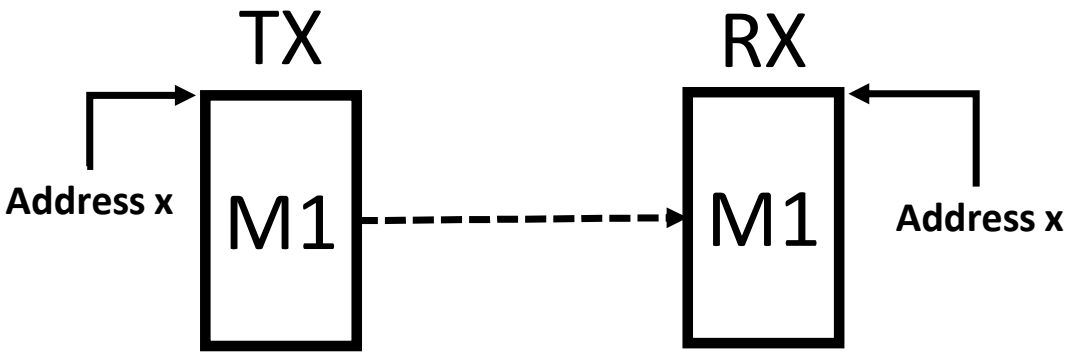
Parallelized Queues



Pack Small Messages



Shared Address Space

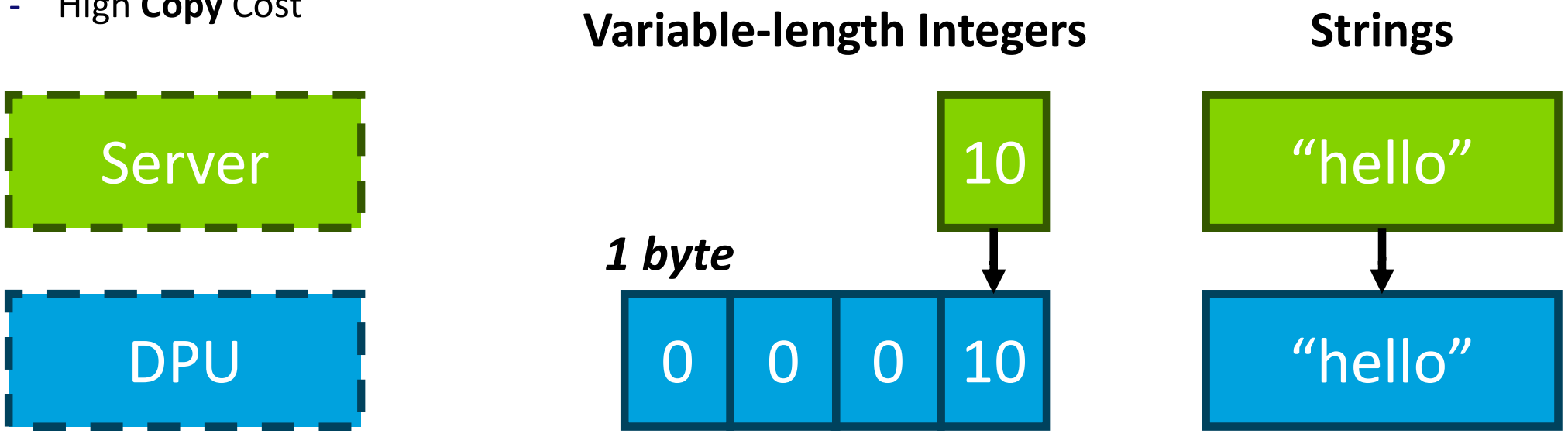


Benchmarking: Deserialization Workload Types

Discriminate two types of costs:

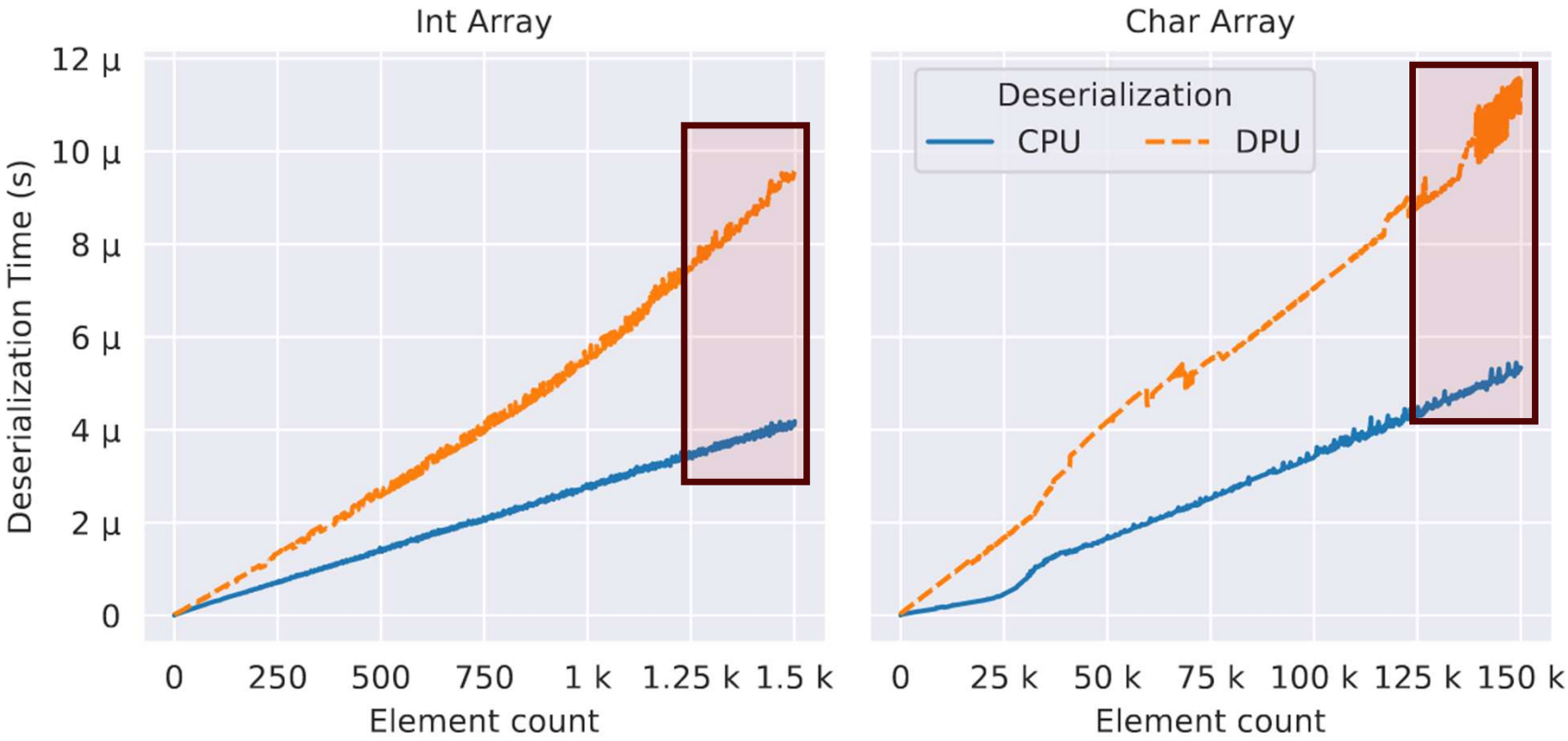
- High **Compute** Cost
- High **Copy** Cost

Compressed Bytes	Max. Value	Generated Proportion
1	127	50%
2	16383	25%
3	2 millions	12.5%
4	2 billions	0.075%
5	max	0.05%

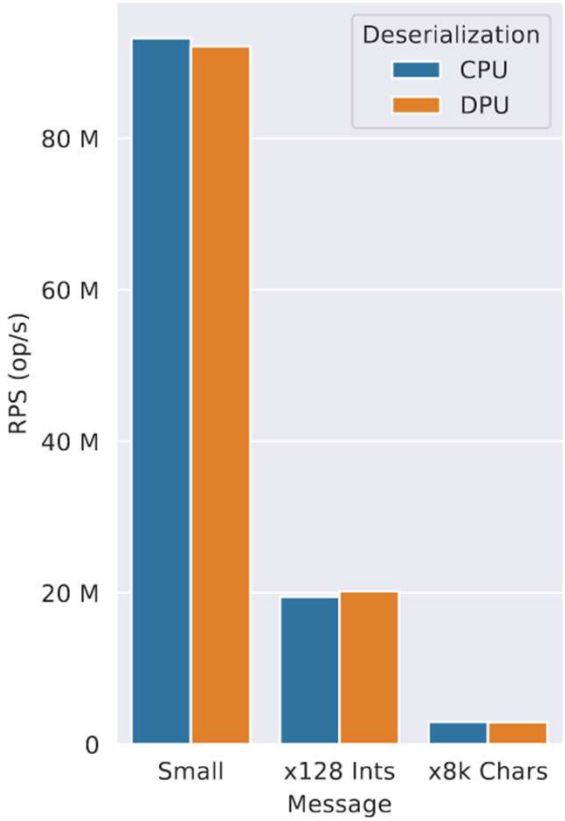


Benchmarking Deserialization

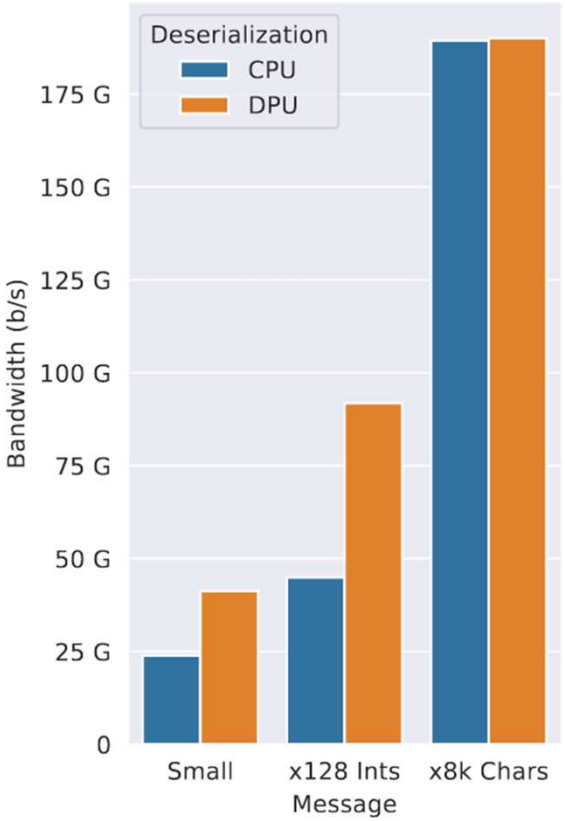
2 DPU cores match the performance of 1 CPU Cores



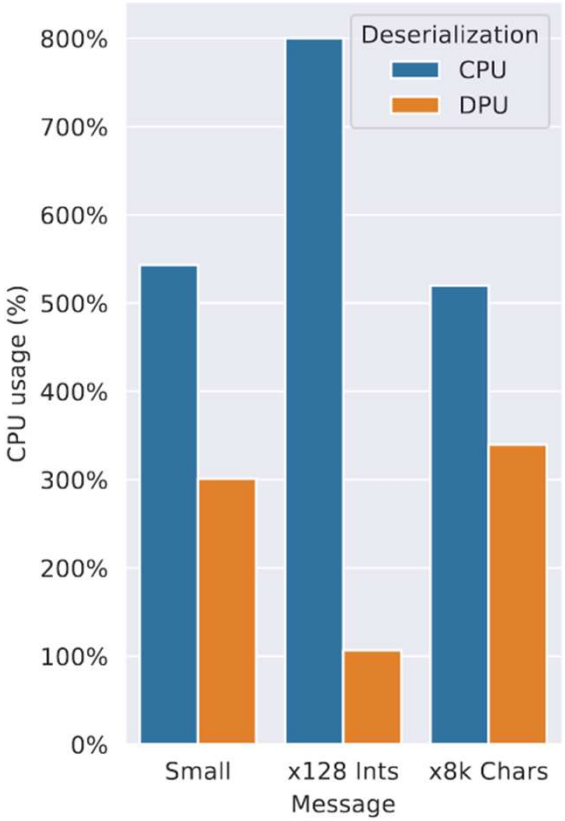
Benchmarking Deserialization in the RPC Datapath



Requests/Sec

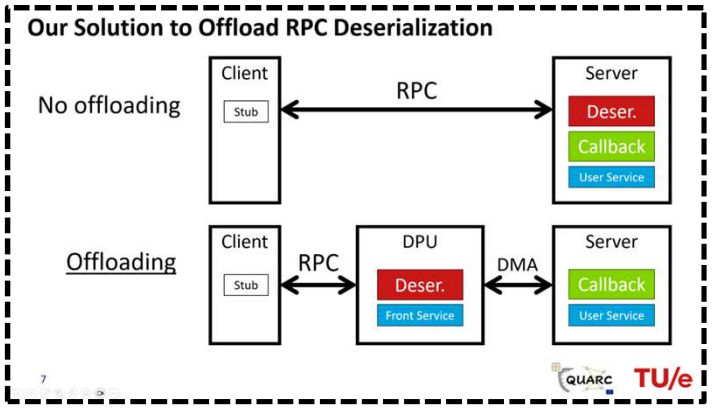


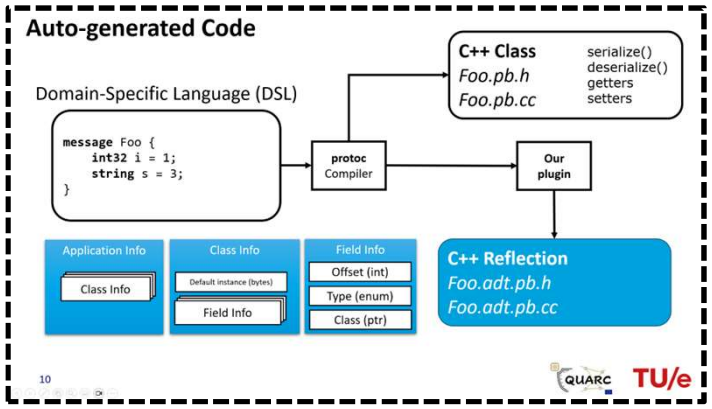
Bandwidth (PCIe)

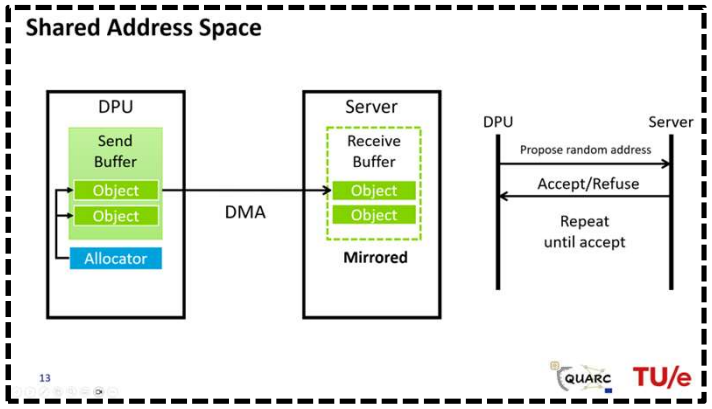


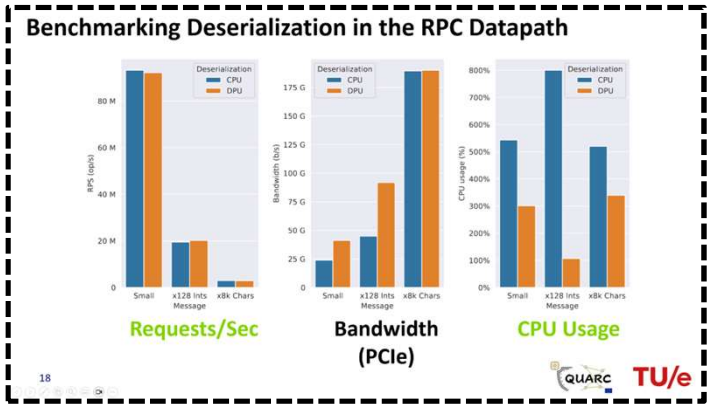
CPU Usage

Conclusion





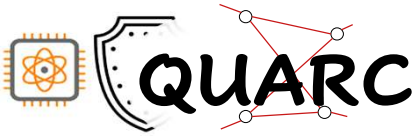




Repository

r.r.a.frantz@tue.nl

This work was partly funded by the European Union Horizon Europe grant number 101073355, and by the grant PID2021-123041OB-I00 funded by MCIN/AEI/10.13039/501100011033 and by ”ERDF A way of making Europe”.



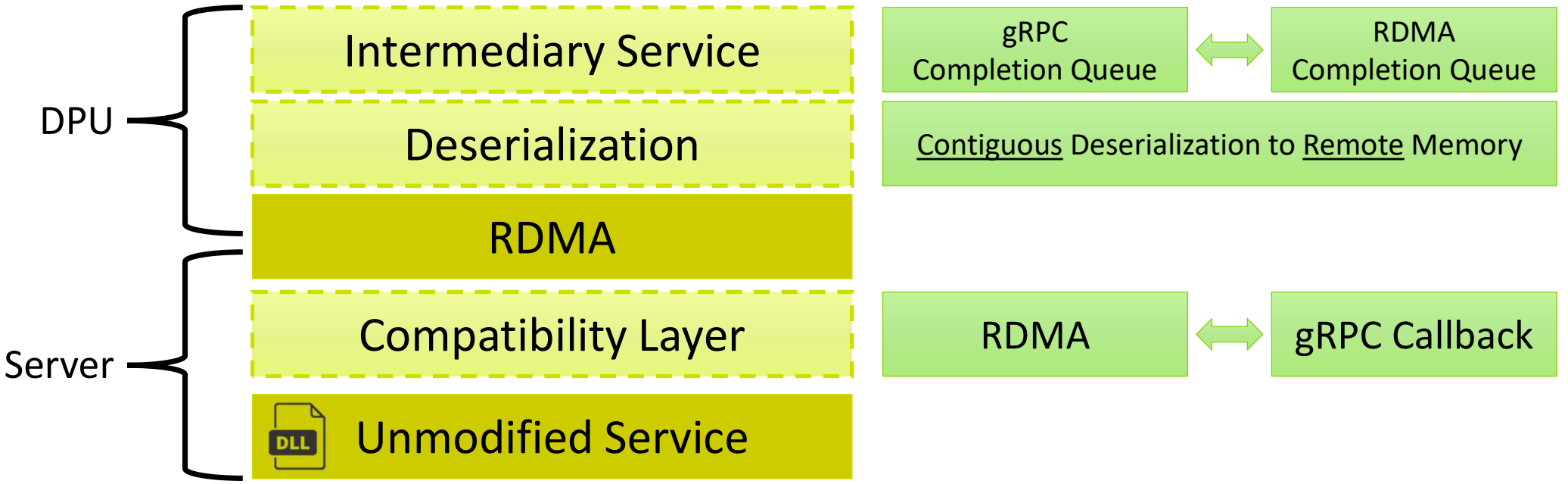
Questions

Protocol Buffer DPU Offloading in the RPC Datapath

Raphaël Frantz – r.r.a.frantz@tue.nl

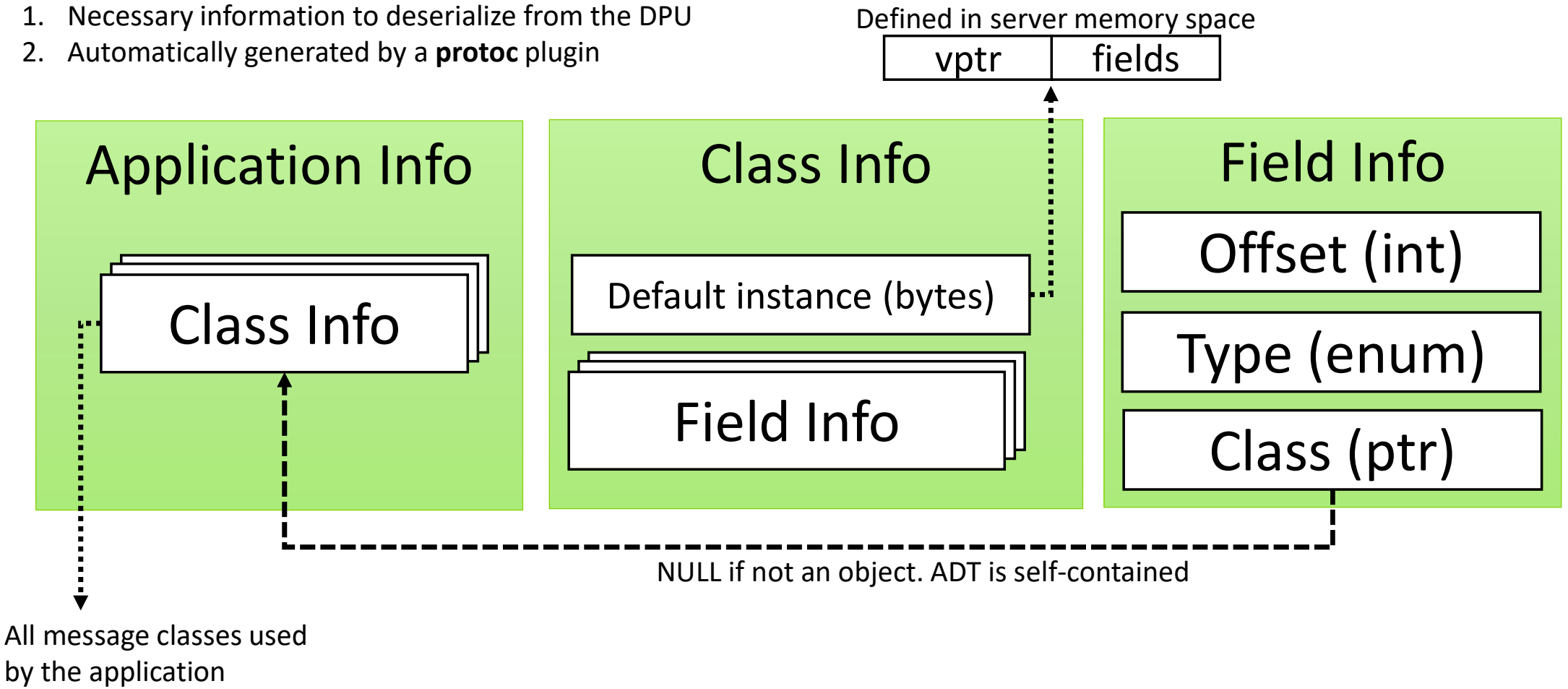
Integrate the proposed Solution with gRPC

Implement once



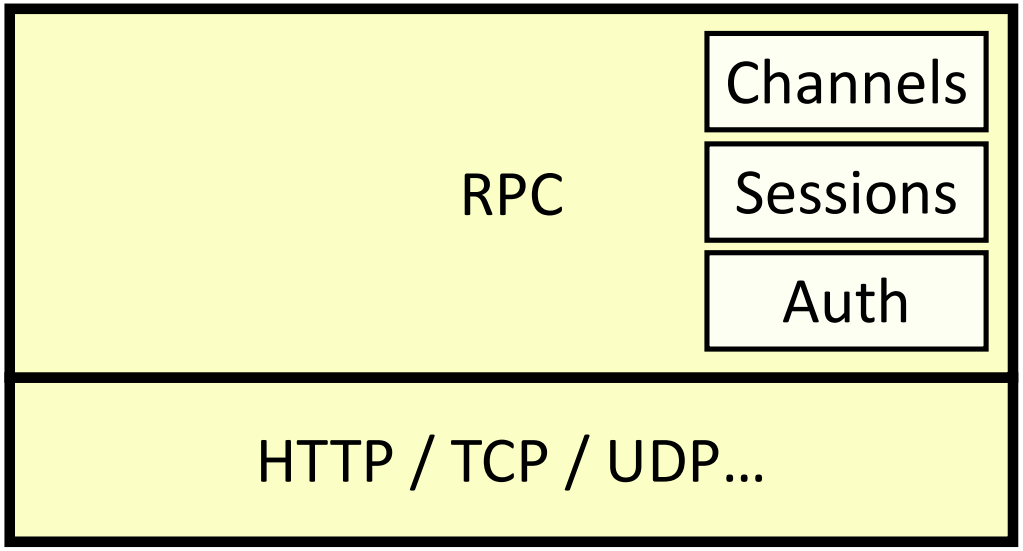
Accelerator Description Table (ADT)

1. Necessary information to deserialize from the DPU
2. Automatically generated by a **protoc** plugin



How to communicate between services?

1. **Slow** to develop: TCP, HTTP…
2. **Fast** to develop: Remote Procedure Calls (RPCs)
   - Client/Server → Function call/Callback



client code

```
client.login();
```

server code

```
login() { … }
```

Offloading RPC Deser. – RDMA

1. *Kernel Bypass*
   - High bandwidth
   - Low latency
2. Competing Standards
   - Infiniband
   - RoCE
   - iWarp
3. Common API
   - libibverbs