

Using ØMQ to implement Communication Middleware

Wojciech Sliwinski, BE-CSS, CERN, Geneva

`Wojciech.Sliwinski@cern.ch`
`https://www.linkedin.com/in/wojciechsliwinski/`

Based on talk of **Joel Lauener** @ ICALEPCS 2017, Barcelona

“**Streaming Readout X**” workshop, Jefferson Lab
18th May 2022



ØMQ

- Networking library & concurrency framework
- Simple socket style API
- Supports inter-thread, inter-process and inter-host
- Provides several socket patterns
- Fast & scalable
- Open source (currently LGPLv3 but moving to MPLV2)

A. Dworak *et al.*, "Middleware trends and market leaders 2011", ICALEPCS 2011
<http://cern.ch/go/G9RC>

RDA - Remote Device Access

RDA2



<https://www.synthesis.co.za/agile-software-architecture-super-agile/>

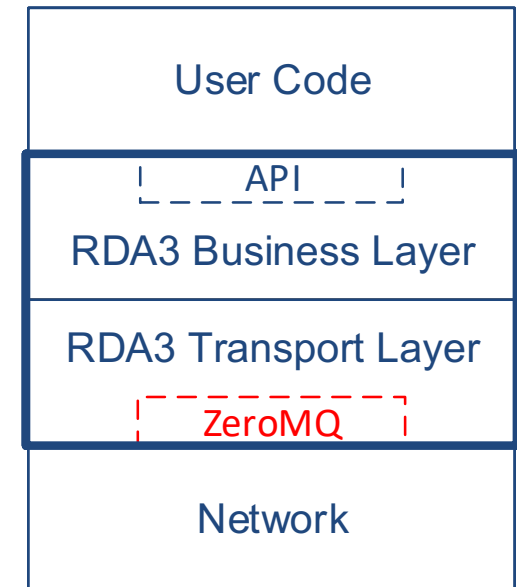
RDA3



Slim Thug f/ Paul Wall "Top Drop" (2009)

RDA3 architectural overview

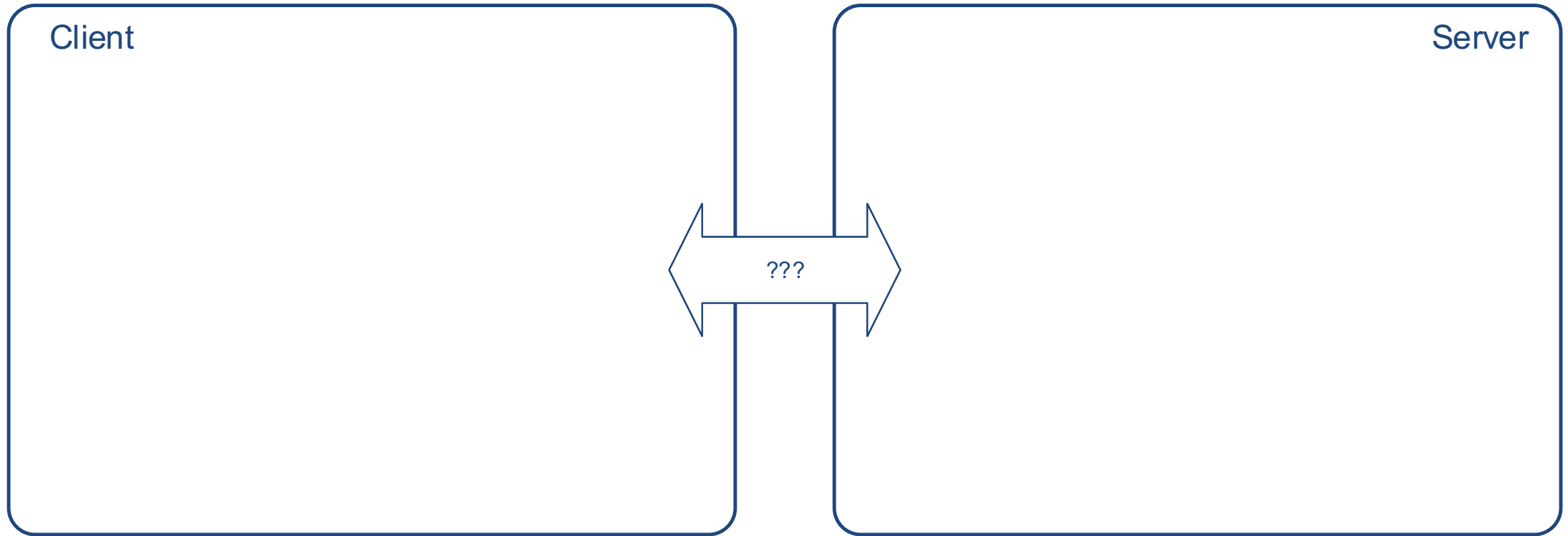
- **Transport Layer**
 - Abstraction of underlying network library
- **Business Layer**
 - Agnostic of network library
 - Implements the device-property model
 - Request/Reply: *Get* (read) and *Set* (write)
 - Publish/Subscribe



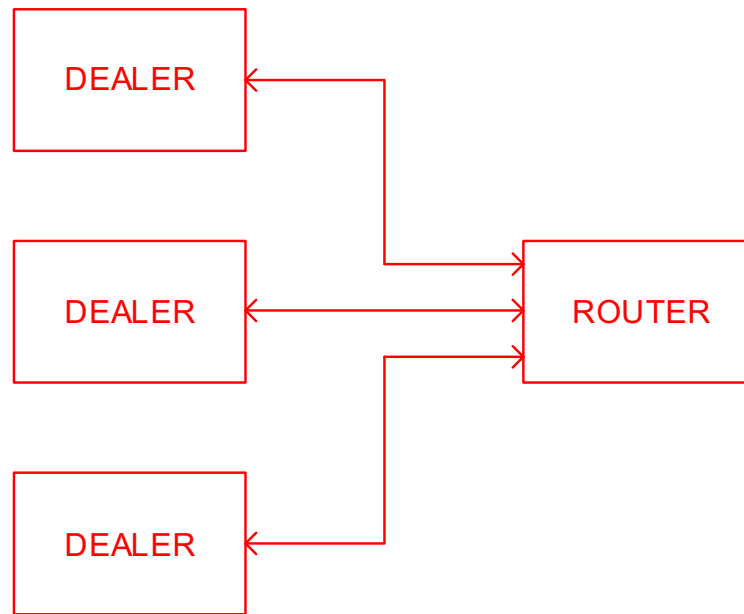
N. Trofimov *et al.*, “Remote Device Access in the new CERN Accelerator Controls middleware”,
ICALEPCS 2001
<http://cern.ch/go/9MSk>

Client-server communication

- RDA3
- User
- ZeroMQ

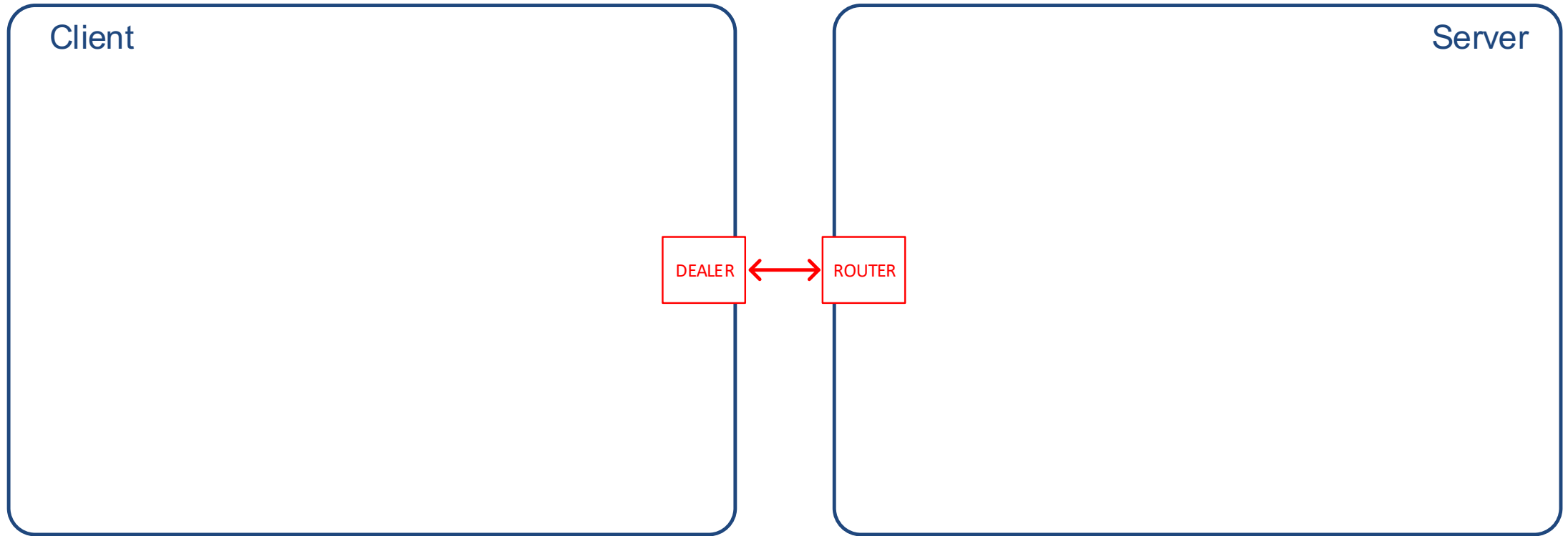


ZeroMQ DEALER/ROUTER pattern



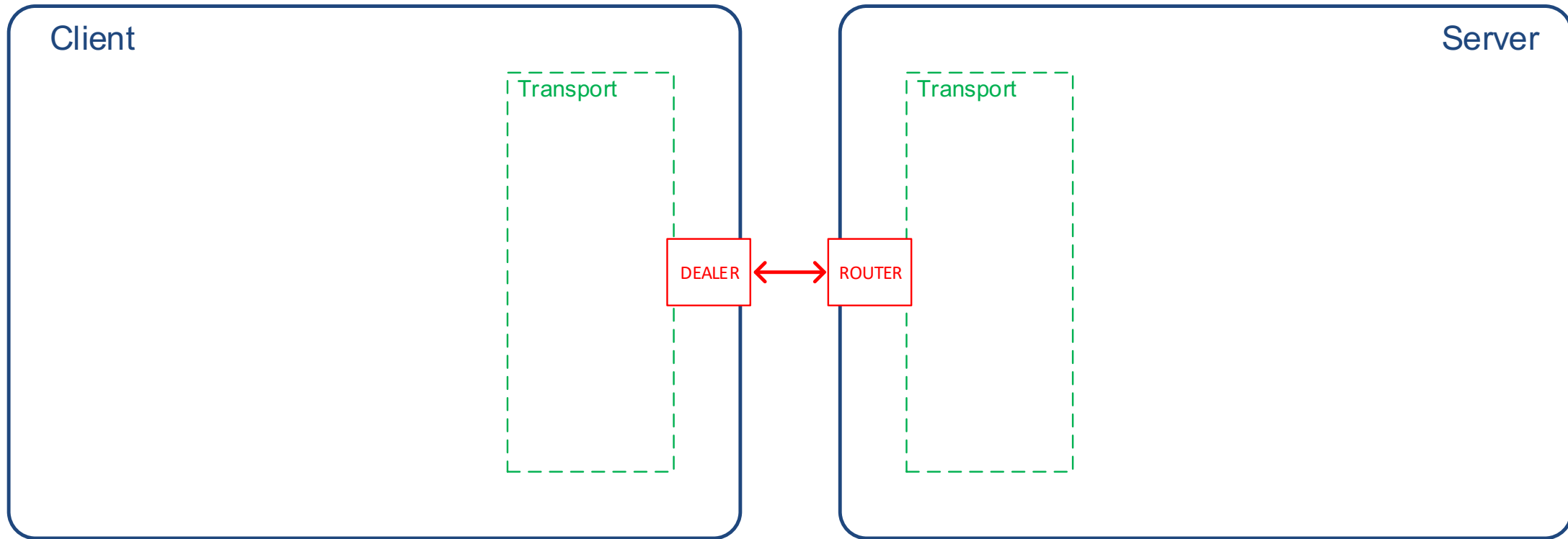
Client-server communication

- RDA3
- User
- ZeroMQ



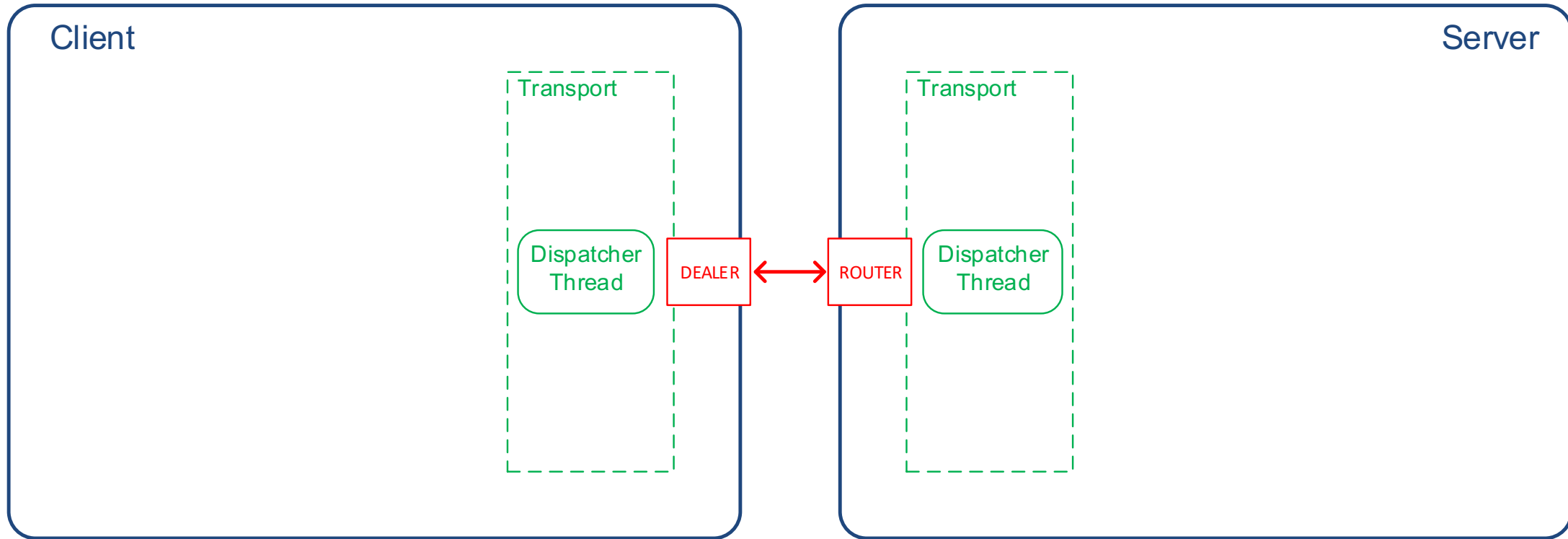
Transport layer

- RDA3
- User
- ZeroMQ



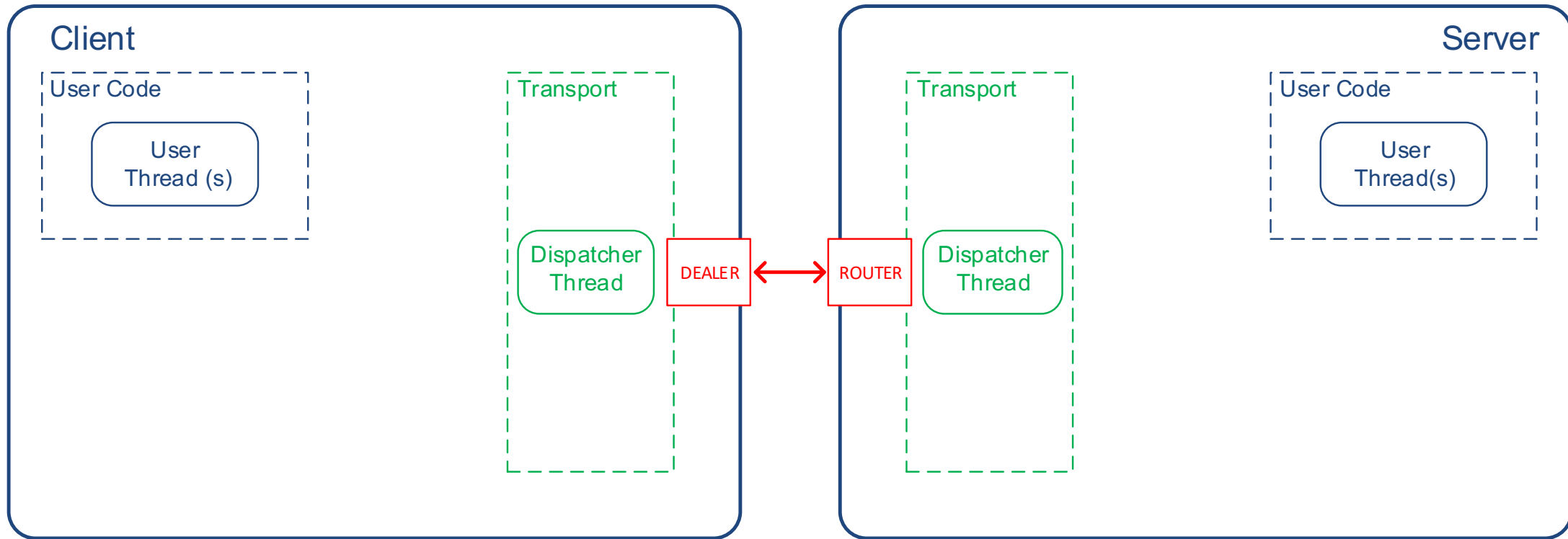
Transport layer

- RDA3
- User
- ZeroMQ



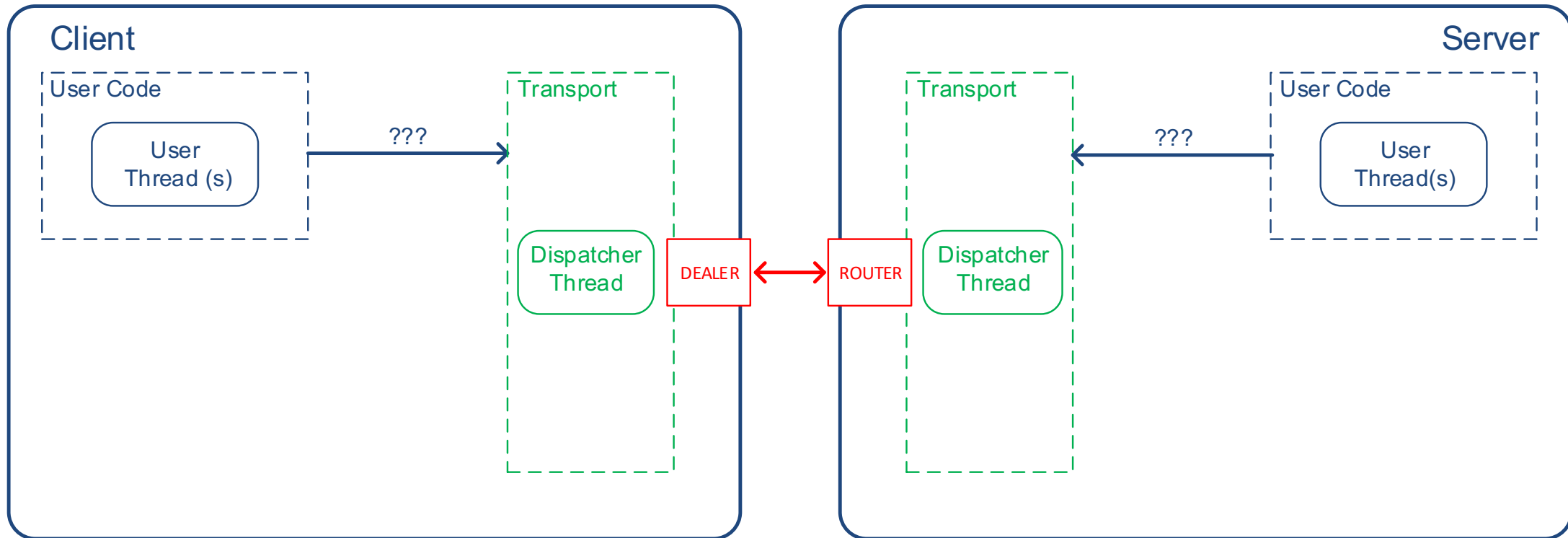
User layer

- RDA3
- User
- ZeroMQ

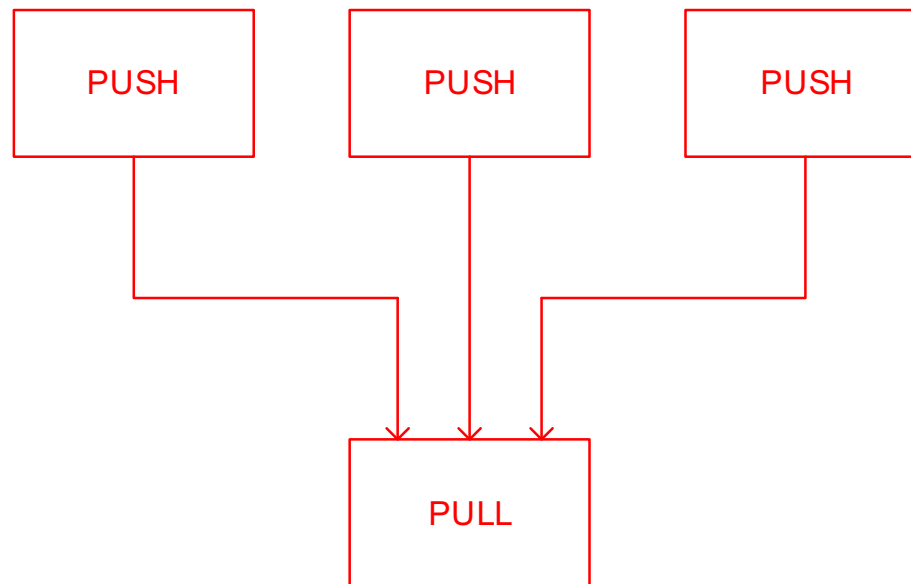


User layer

- RDA3
- User
- ZeroMQ

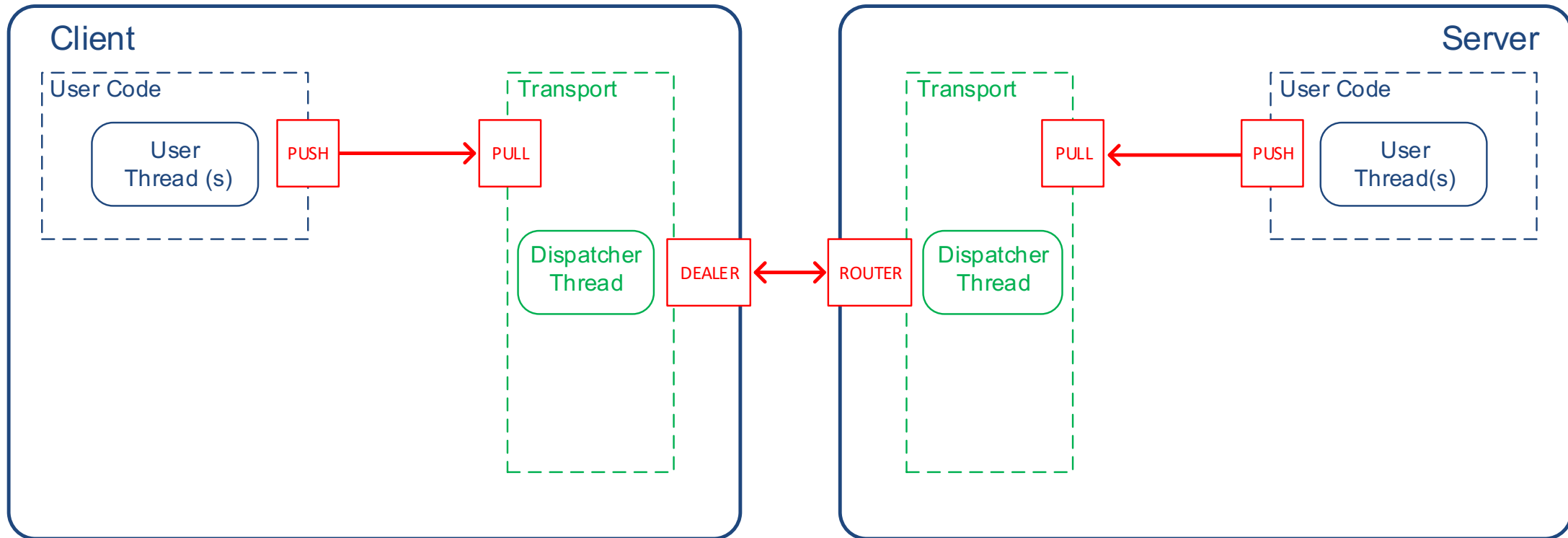


ZeroMQ PUSH/PULL pattern



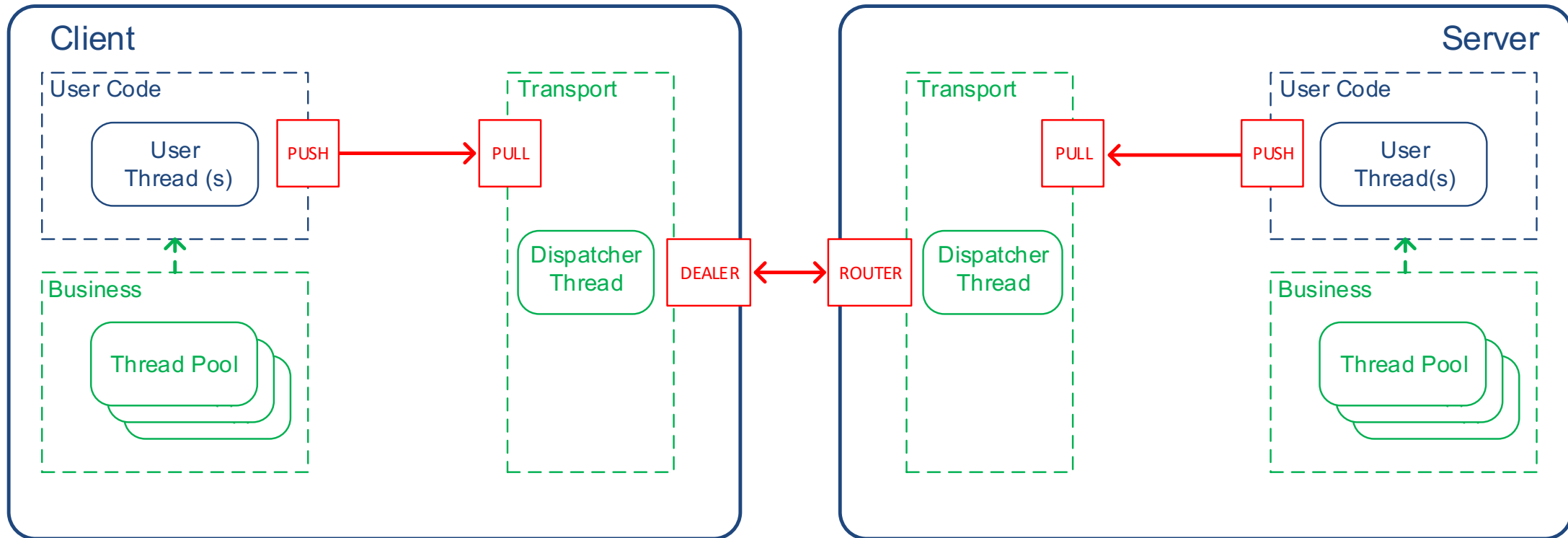
User layer

- RDA3
- User
- ZeroMQ



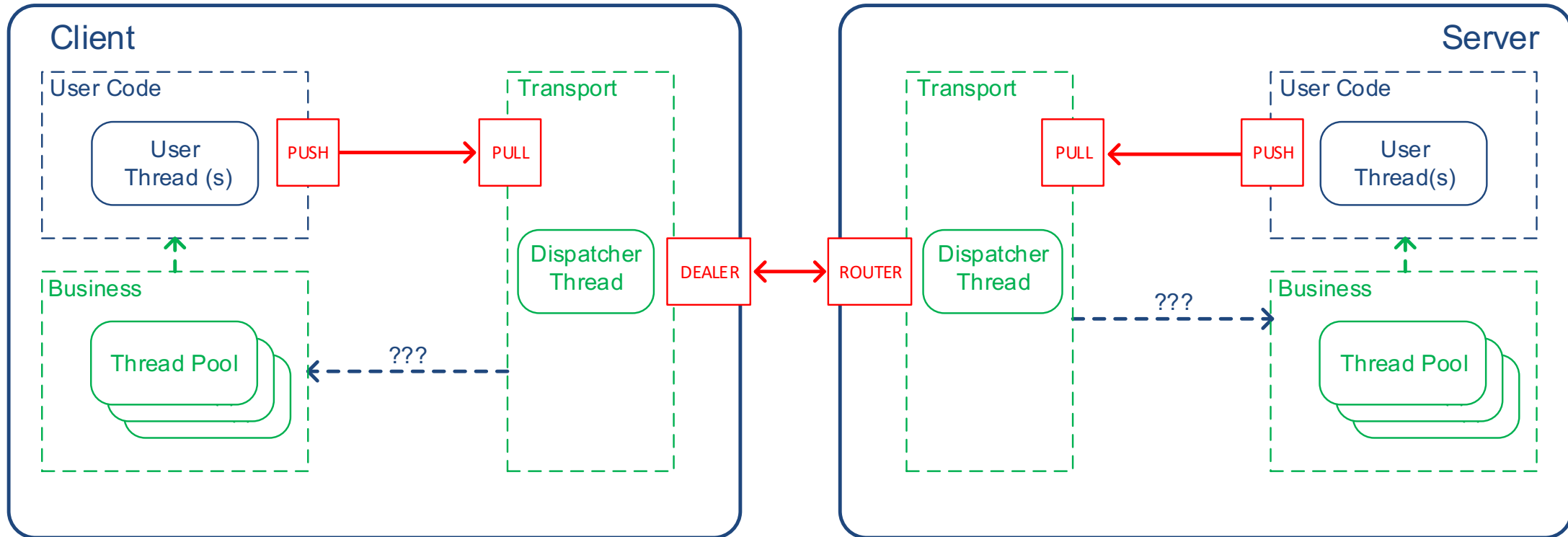
Business layer

- RDA3
- User
- ZeroMQ



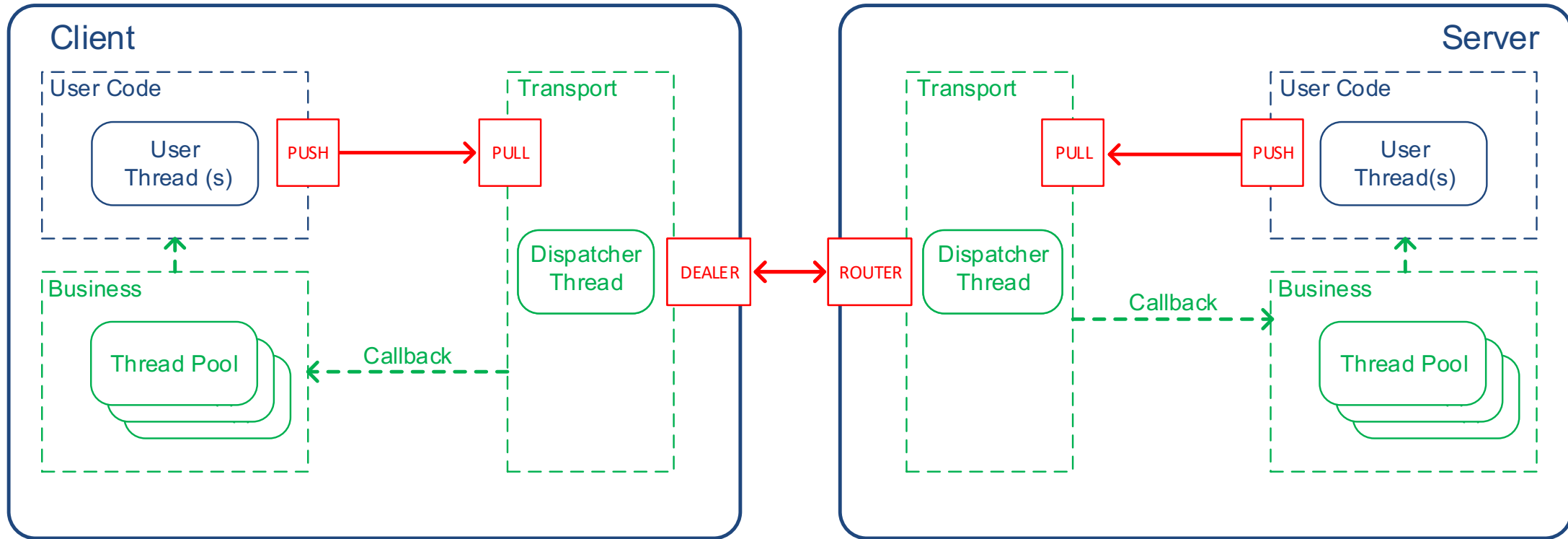
Business layer

- RDA3
- User
- ZeroMQ



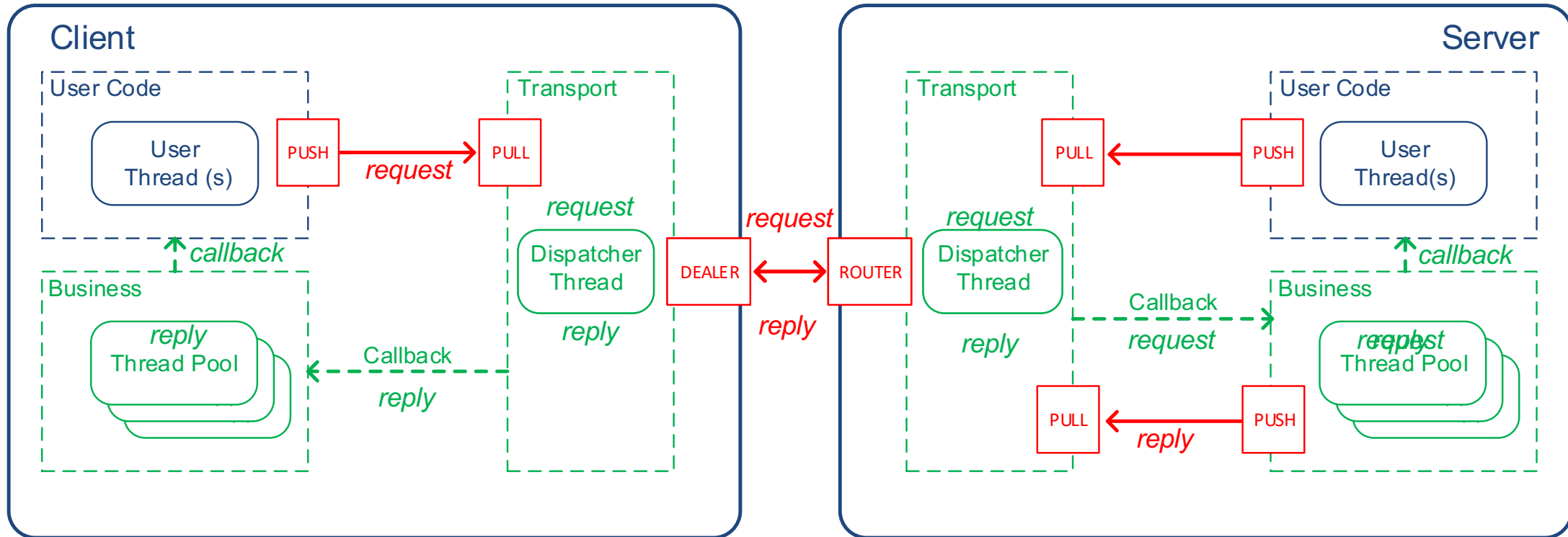
Business layer

- RDA3
- User
- ZeroMQ



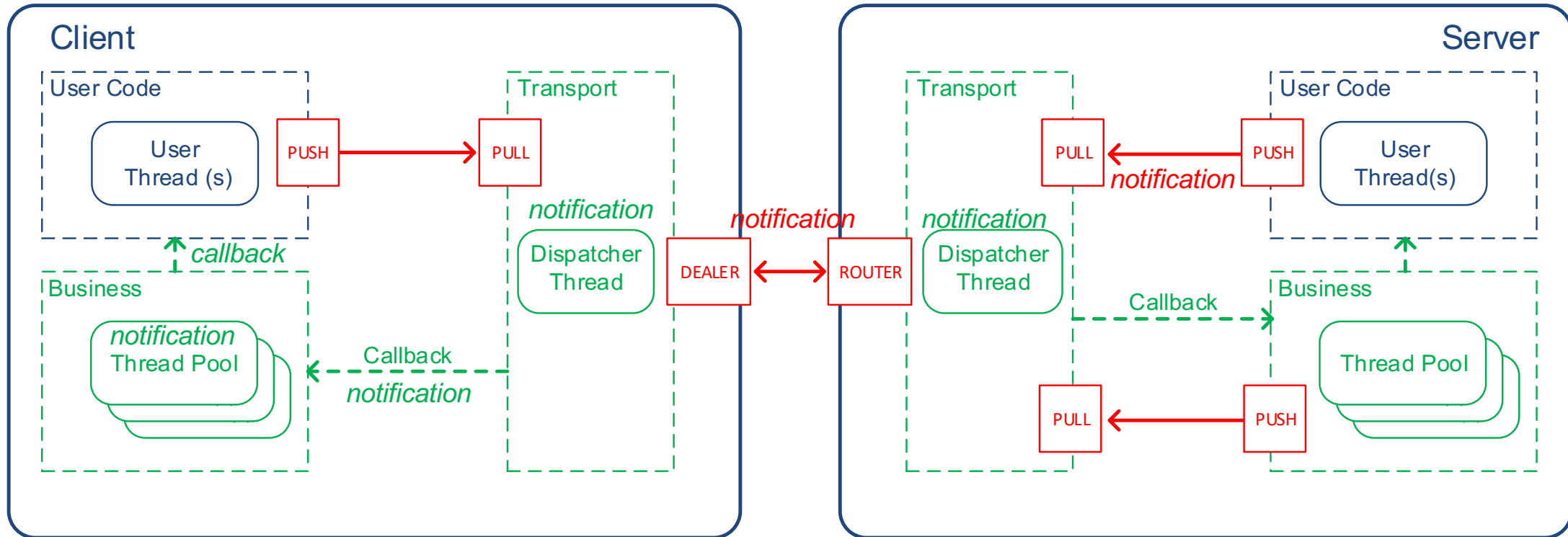
Request/Reply (e.g. *Get* call)

- RDA3
- User
- ZeroMQ



Publish/Subscribe: Sending a notification

- RDA3
- User
- ZeroMQ



Our experience with ØMQ

Pros	Cons or “good to know”
Proposed architecture proved to be efficient	Lack of built-in heart-beating mechanism for connection management (2013)
Solid, stable, high quality networking library	ZMQ Socket’s HWM (High-Water-Mark) policy for max. queue size based on message count not sufficient. We also need max. queue size in bytes .
Outstanding scalability & reliability Async, non-blocking communication is a “game changer”	Lack of backpressure mechanism for publishers in case of slow-consumers
Portfolio of different socket communication patterns	Lack of timeout control as communication is async
Active & responsive community	Single-thread access to ZMQ socket for dispatching messages
Excellent online documentation	Java: JNI (jzmq) & pure-Java (jeromq) not equal feature-wise

If we developed RDA3 again today ...

- ZeroMQ would be still one of our **top choices**
- However, we would like to use “**web friendly**” protocol
 - HTTP/2 for transport
 - Non-blocking communication
 - Bi-directional streaming (e.g. WebSocket, RSocket)
 - Cluster/Cloud friendly (Kubernetes, e.g. gRPC)
- Replace in-house serialization (text, binary) with an **industry standard**
 - Protocol Buffers, Apache Avro, MessagePack, ...

Conclusions

- ZeroMQ is used in operation for all CERN accelerators for 7 years
 - **Very good** operational experience
 - **Outstanding** scalability & reliability in **peer-to-peer** communication
 - **Missing** Cluster/Cloud capabilities (not ready for Kubernetes)
- RDA3 (based on ZeroMQ) was developed in **collaboration** with GSI
 - Already used @ GSI; will be used for the new FAIR complex
- Migrating to ZeroMQ was **the right decision**
- **Development process** was a key success factor
 - Scrum-like, short iterations, peer reviews, test early, CI/CD, deliver often

`Wojciech.Sliwinski@cern.ch`
`https://cmwdoc.web.cern.ch/`