Interoperability in Enterprise Hyperledger Fabric Blockchains

8th Distributed Ledger Technology Workshop (DLT'26)

Luca Olivieri[1], Aradhita Mukherjee[2], Nabendu Chaki[3], Agostino Cortesi[1]

1 Ca' Foscari University of Venice, Venice, Italy
2 Netaji Subhash Engineering College, Kolkata, India
3 University of Calcutta, Kolkata, India

Pula (Cagliari), Italy - June 4-6, 2026

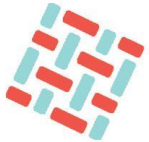
Finanziato dall'Unione europea NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani PIANO NAZIONALE DI RIPRESA E RESILIENZA
SERICS SECURITY AND RIGHTS IN THE CYBERSPACE

Hyperledger Fabric

Most popular for enterprise-grade **permissioned** blockchains:

- Powered by **Linux Foundation** (LF Decentralized Trust)
- Supported by **major cloud providers** (AWS, IBM, Oracle, …)

Hyperledger Fabric



HYPERLEDGER FABRIC

Most popular for enterprise-grade **permissioned** blockchains:

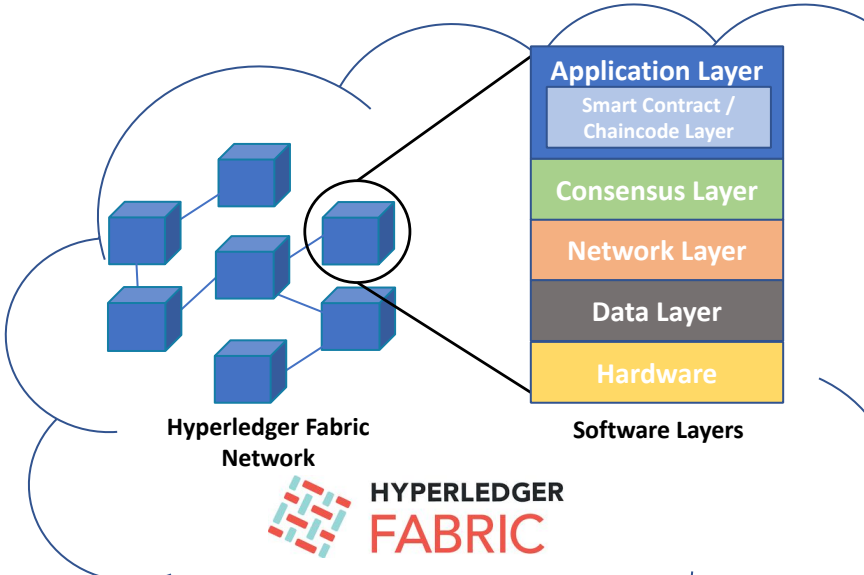
- Powered by **Linux Foundation** (LF Decentralized Trust)
- Supported by **major cloud providers** (AWS, IBM, Oracle, …)

Smart contracts:

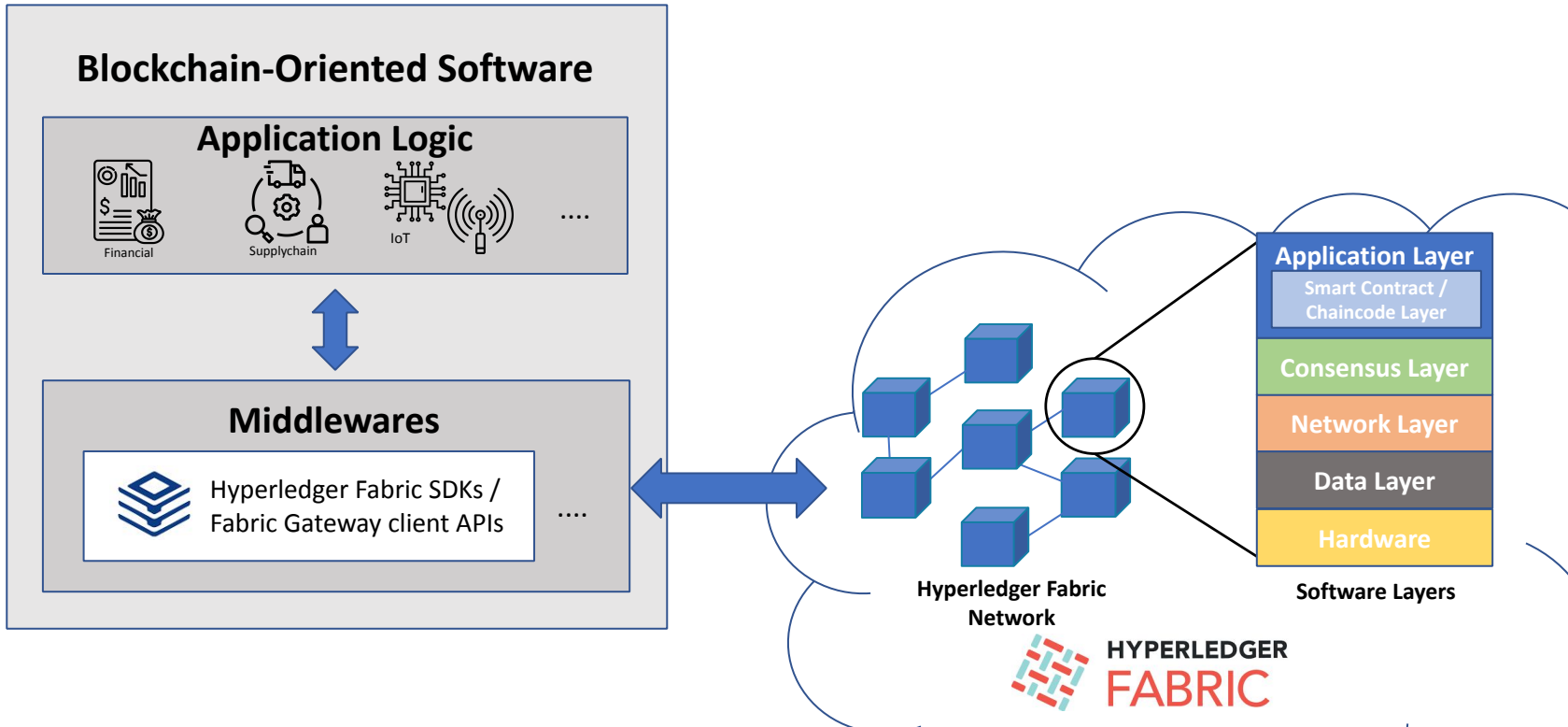
- *Chaincode* logic
- Written in **general-purpose languages** [1]

[1] L. Olivieri *et al*., "General-Purpose Languages for Blockchain Smart Contracts Development: A Comprehensive Study," in *IEEE Access, 2024*

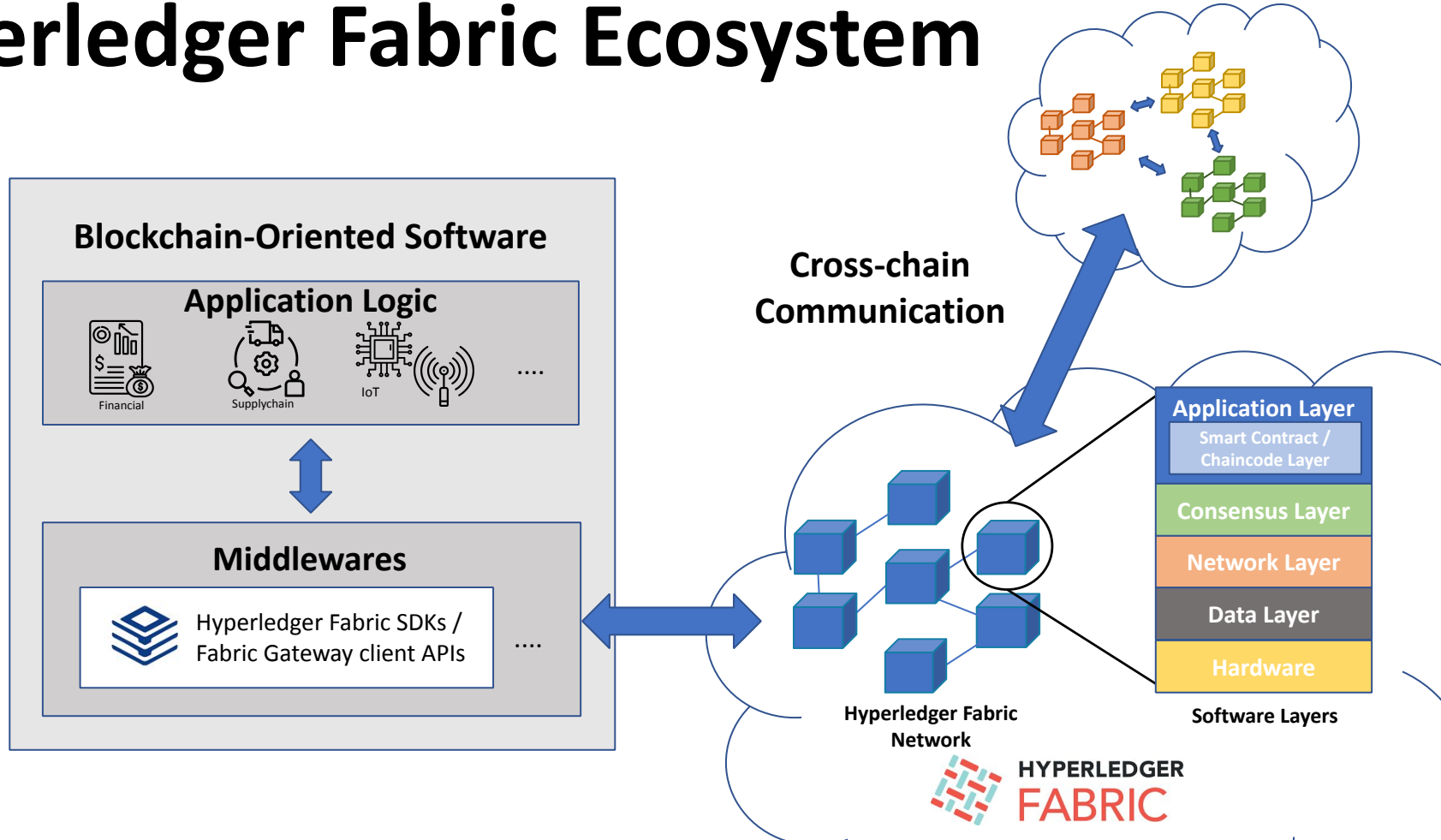
Hyperledger Fabric Ecosystem



Hyperledger Fabric Ecosystem

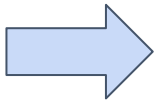


Hyperledger Fabric Ecosystem



How to Achieve Interoperability

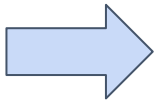
- Transaction Request/Response
- Event Emission



Inter-framework
Common also in other blockchains

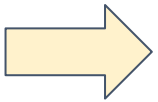
How to Achieve Interoperability

- Transaction Request/Response
- Event Emission



Inter-framework
Common also in other blockchains

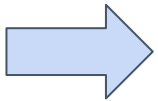
- Channels Communications
- Private Data Collections



Intra-framework
Hyperledger Fabric native solutions

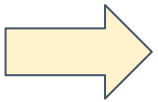
How to Achieve Interoperability

- Transaction Request/Response
- Event Emission



Inter-framework
Common also in other blockchains

- Channels Communications
- Private Data Collections



Intra-framework
Hyperledger Fabric native solutions

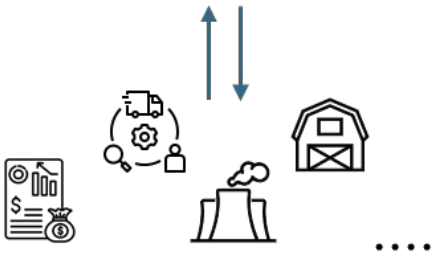
Others

- Custom and non-conventional solutions
- Third-party frameworks

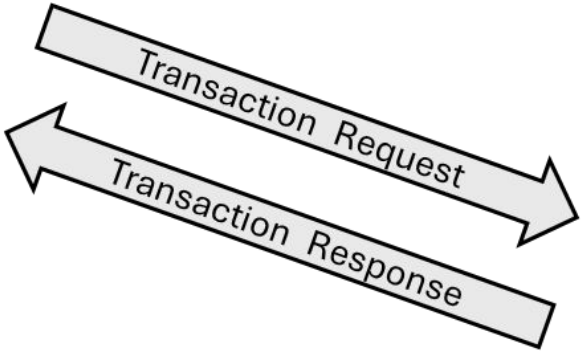
Inter-framework Interoperability

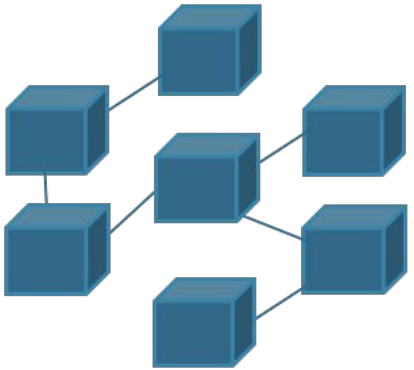
Transaction Request/Response

Transaction Sender



External Systems





Blockchain

Pro and Cons

- ✓ **Primary way to interact** with the blockchain network
- ✓ **External** communications
- ✓ Interoperability **inter-framework**
- ✓ **Traceability** (clear evidence and track of interactions)
- ✓ **Simplicity** (easy to understand and implement)

Pro and Cons

- ✓ **Primary way to interact** with the blockchain network
- ✓ **External** communications
- ✓ Interoperability **inter-framework**
- ✓ **Traceability** (clear evidence and track of interactions)
- ✓ **Simplicity** (easy to understand and implement)

- − **Synchronous and Higher Latency** (require to wait for a response)
- − **Scalability Limitations** (bottlenecks and cost increases)
- − **Error handling** (transaction and smart contracts)

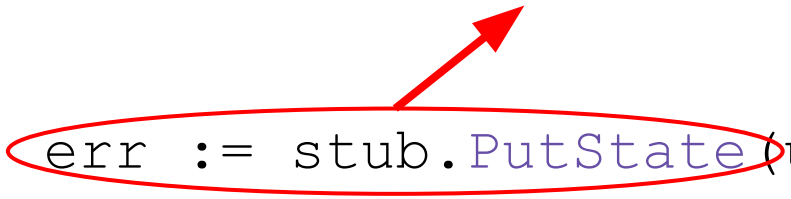
Error handling in HF smart contracts

```
err := stub.PutState(user, updatedBalance)
return shim.Success("Cross-chain transfer performed.")
```

L. Olivieri and L. Negrini, "Don’t Panic: Error Handling Patterns in Go Smart Contracts and Blockchain Software," *2025 7th Conference on Blockchain Research & Applications for Innovative Networks and Services (BRAINS)*

Error handling in HF smart contracts

In HF, **read** and **write** operations on the **world state may fail!**

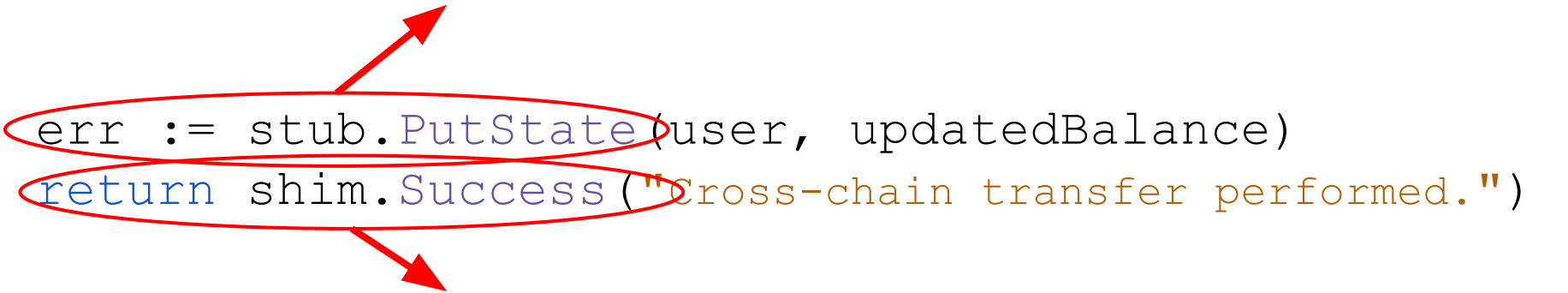


```
err := stub.PutState(user, updatedBalance)
return shim.Success("Cross-chain transfer performed.")
```

L. Olivieri and L. Negrini, "Don’t Panic: Error Handling Patterns in Go Smart Contracts and Blockchain Software," *2025 7th Conference on Blockchain Research & Applications for Innovative Networks and Services (BRAINS)*

Error handling in HF smart contracts

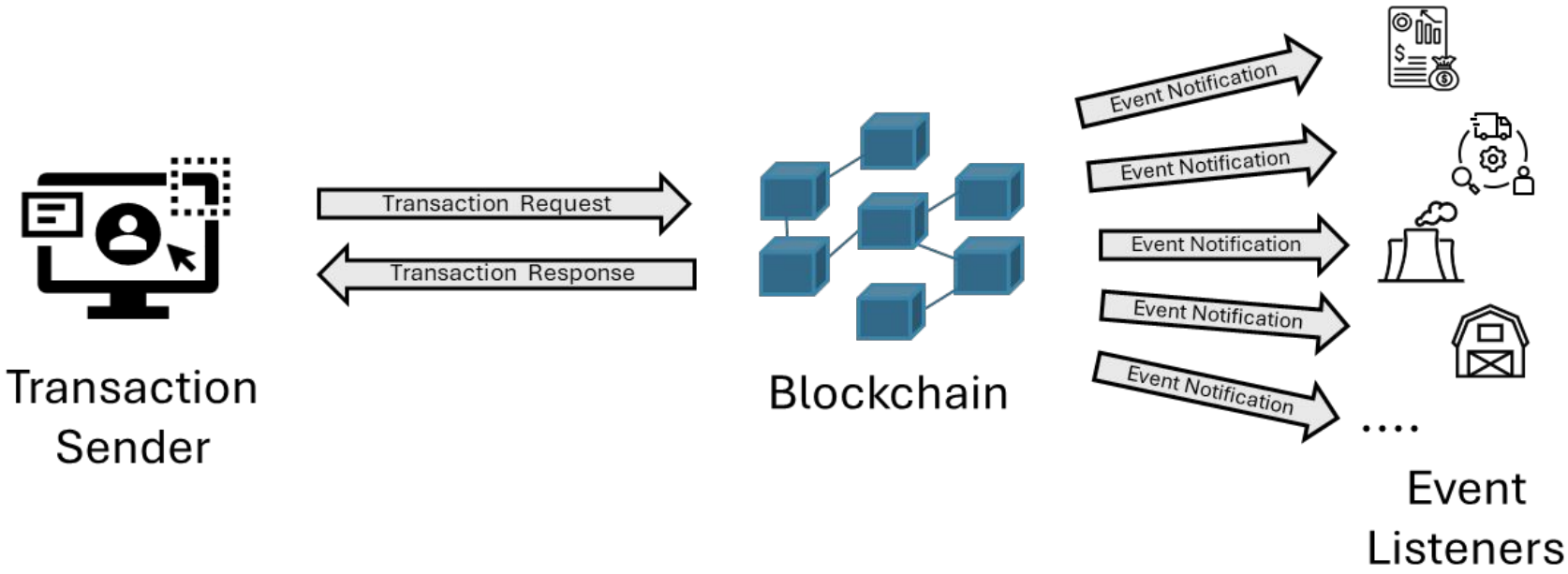
In HF, **read** and **write** operations on the **world state may fail!**



In HF, the **transaction response** is typically **independent** from errors. **It may lead to silent failures!**

L. Olivieri and L. Negrini, "Don't Panic: Error Handling Patterns in Go Smart Contracts and Blockchain Software," *2025 7th Conference on Blockchain Research & Applications for Innovative Networks and Services (BRAINS)*

Event Emission



Pro and Cons

- ✓ **External** communications
- ✓ Interoperability **inter-framework**
- ✓ **Asynchronous** (no wait response from someone)
- ✓ **Simplicity** (easy to understand and implement)

- – **Synchronization, monitoring, management**
- – **Error handling, missing notification**
  (transaction and smart contracts)

Issues in HF

```
err := stub.SetEvent("myEvent", payload)
return shim.Success("OK!")
```

Issues in HF

```
err := stub.SetEvent("myEvent", payload)
return shim.Success("OK!")
```

Event may fail!

Issues in HF

```
err := stub.SetEvent("myEvent", payload)
return shim.Success("OK!")
```

Event may fail!

```
err := stub.SetEvent("EventA", payloadA)
// ...
err := stub.SetEvent("EventB", payloadB)
```

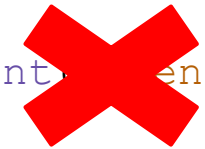
Issues in HF

```
err := stub.SetEvent("myEvent", payload)
return shim.Success("OK!")
```

Event may fail!

Only a single event can be included in a transaction!

```
err := stub.SetEvent("EventA", payloadA)
// ...
err := stub.SetEvent("EventB", payloadB)
```



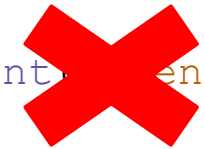
Issues in HF

```
err := stub.SetEvent("myEvent", payload)
return shim.Success("OK!")
```

Event may fail!

Only a single event can be included in a transaction!

```
err := stub.SetEvent("EventA", payloadA)
// ...
err := stub.SetEvent("EventB", payloadB)
```

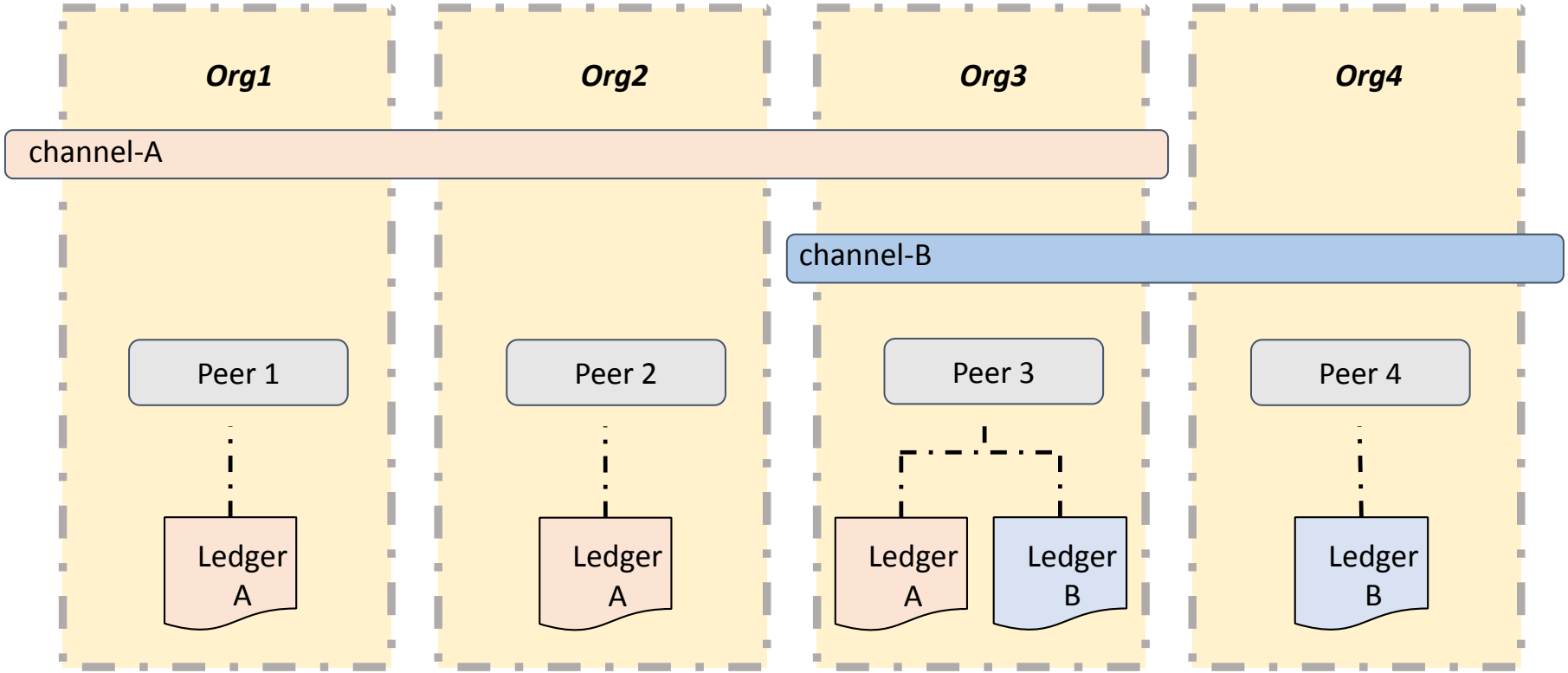


```
stub.InvokeChaincode(...)
// ...
err := stub.SetEvent("evt", payload)
```

Missing notification in intra-framework interoperability

Intra-framework Interoperability

Channels

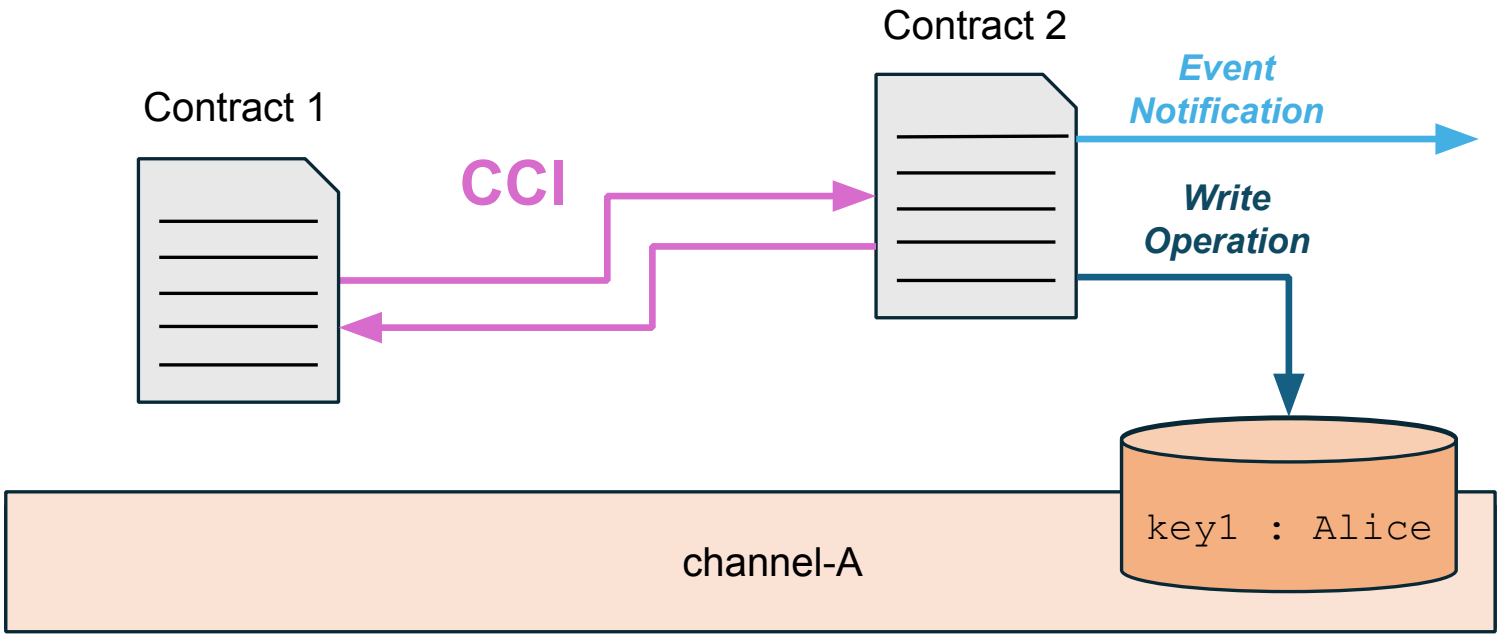

Org1
Org2
Org3
Org4
channel-A
channel-B
Peer 1
Peer 2
Peer 3
Peer 4
Ledger A
Ledger A
Ledger A
Ledger B
Ledger B

Pro and Cons

- ✓ Interoperability **native intra-framework**
- ✓ **Same protocol, same framework, same network**
- ✓ **Confidentiality, privacy preservation between organizations**

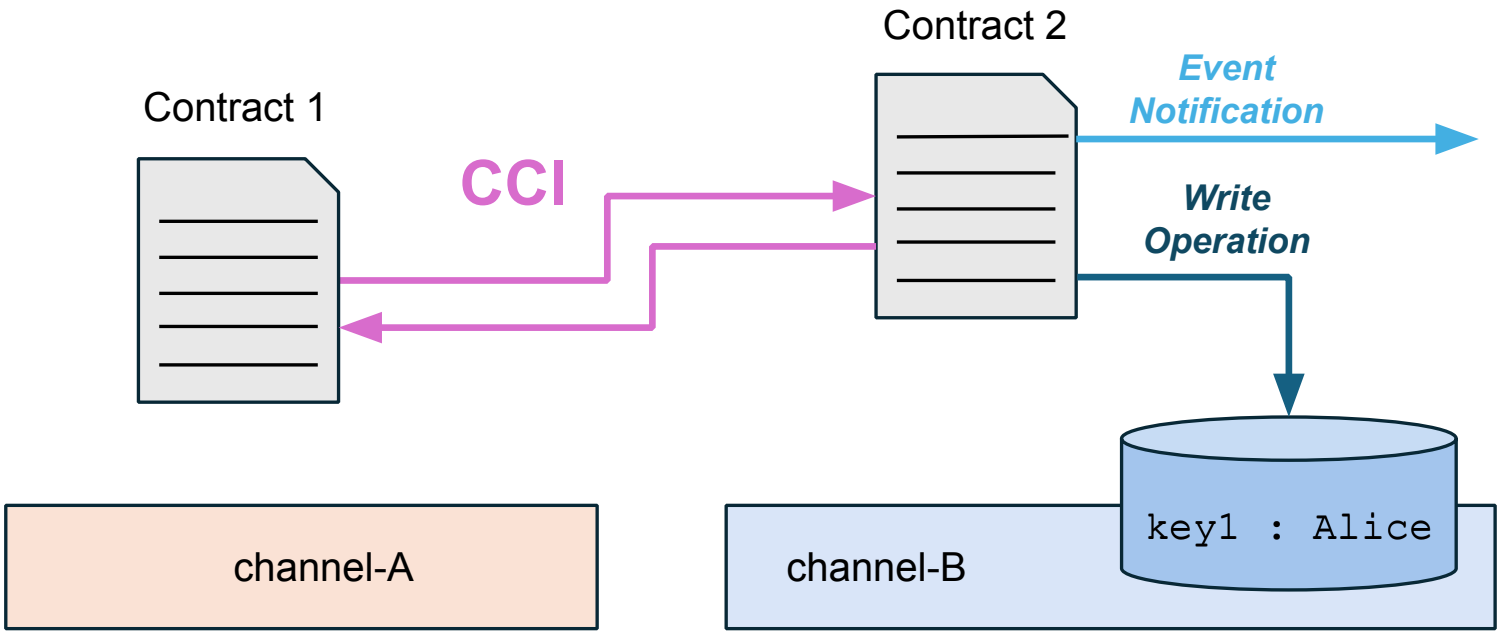
- – **Require system management**
- – **Unsustainable with many organizations**
- – **Lack of traceability and other issues in cross-channel invocations**

Issues in HF



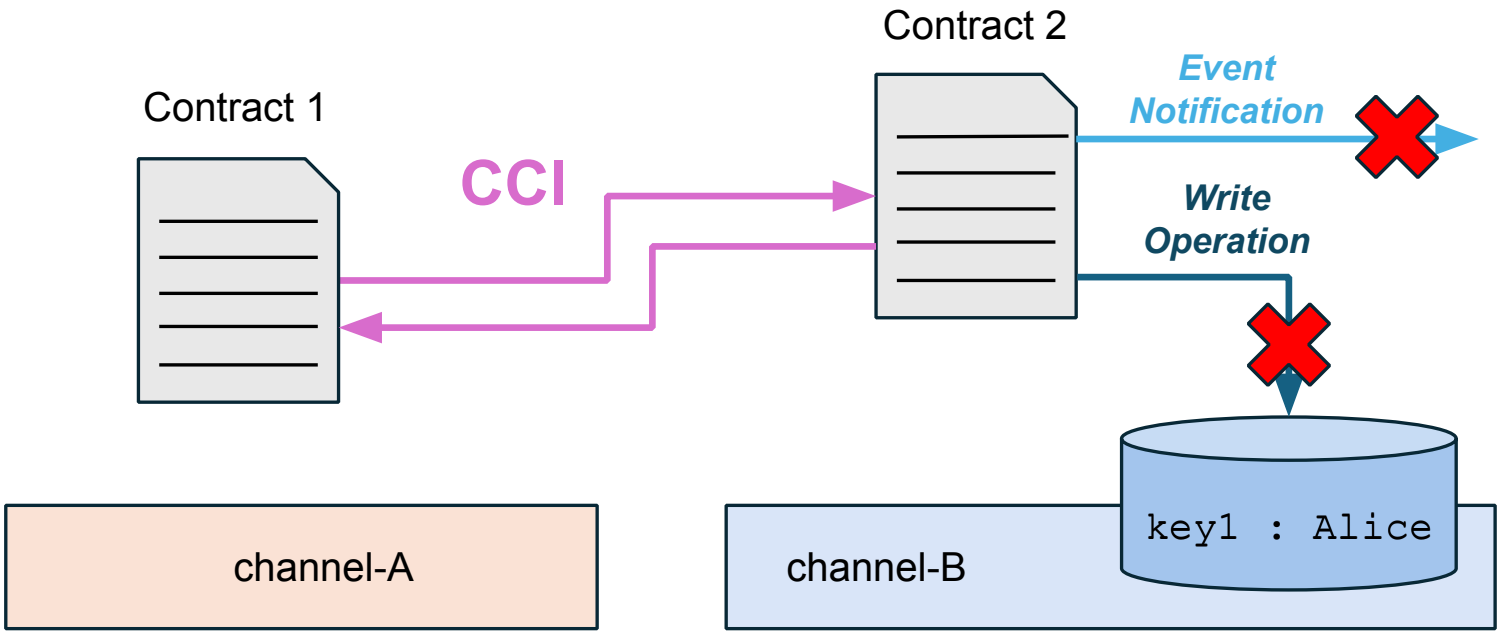
L. Olivieri, "Detection of Cross-Channel Invocation Risks in Hyperledger Fabric," *2025 IEEE 36th International Symposium on Software Reliability Engineering (ISSRE)*

Issues in HF



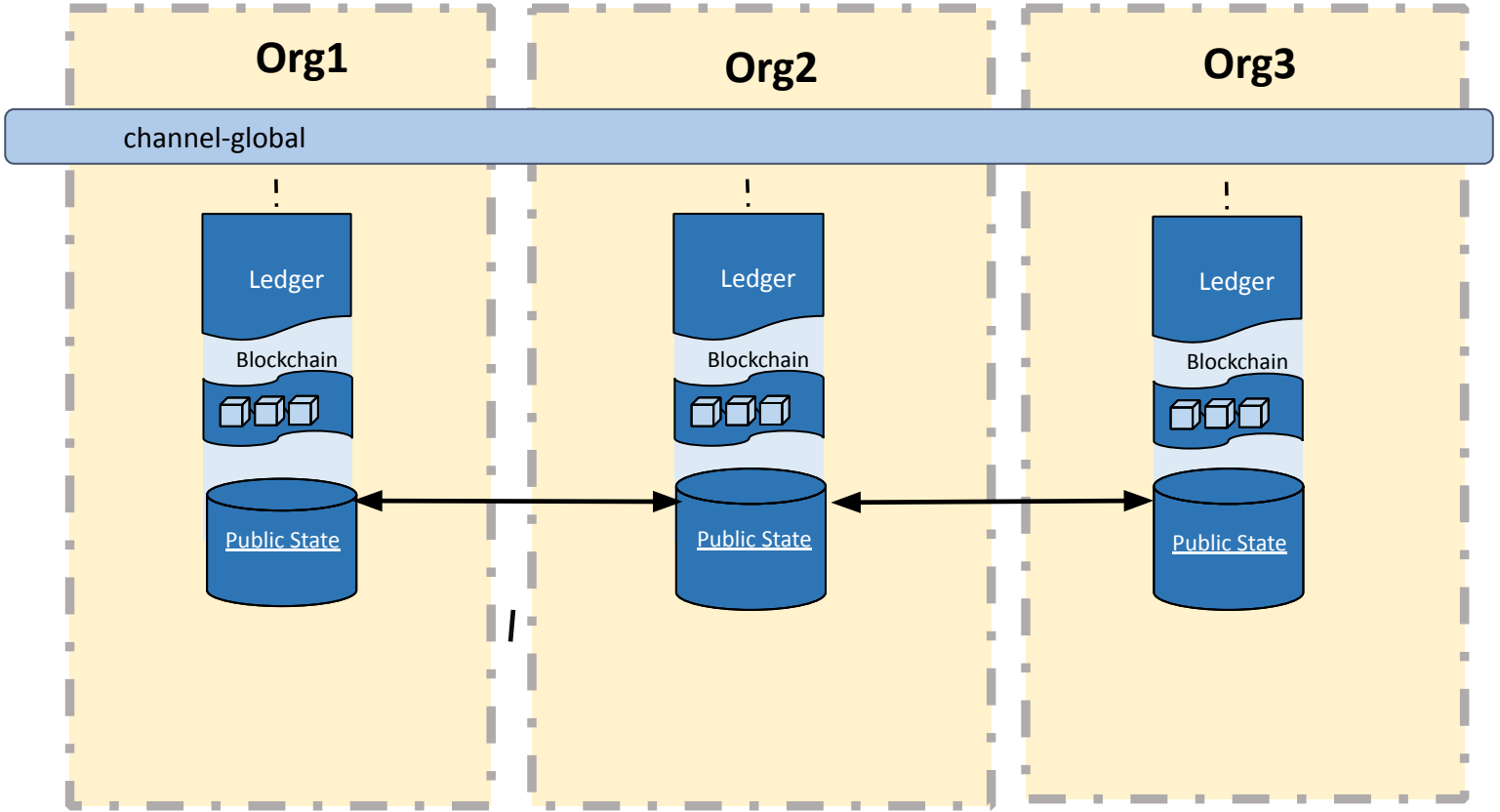
L. Olivieri, "Detection of Cross-Channel Invocation Risks in Hyperledger Fabric," *2025 IEEE 36th International Symposium on Software Reliability Engineering (ISSRE)*

Issues in HF

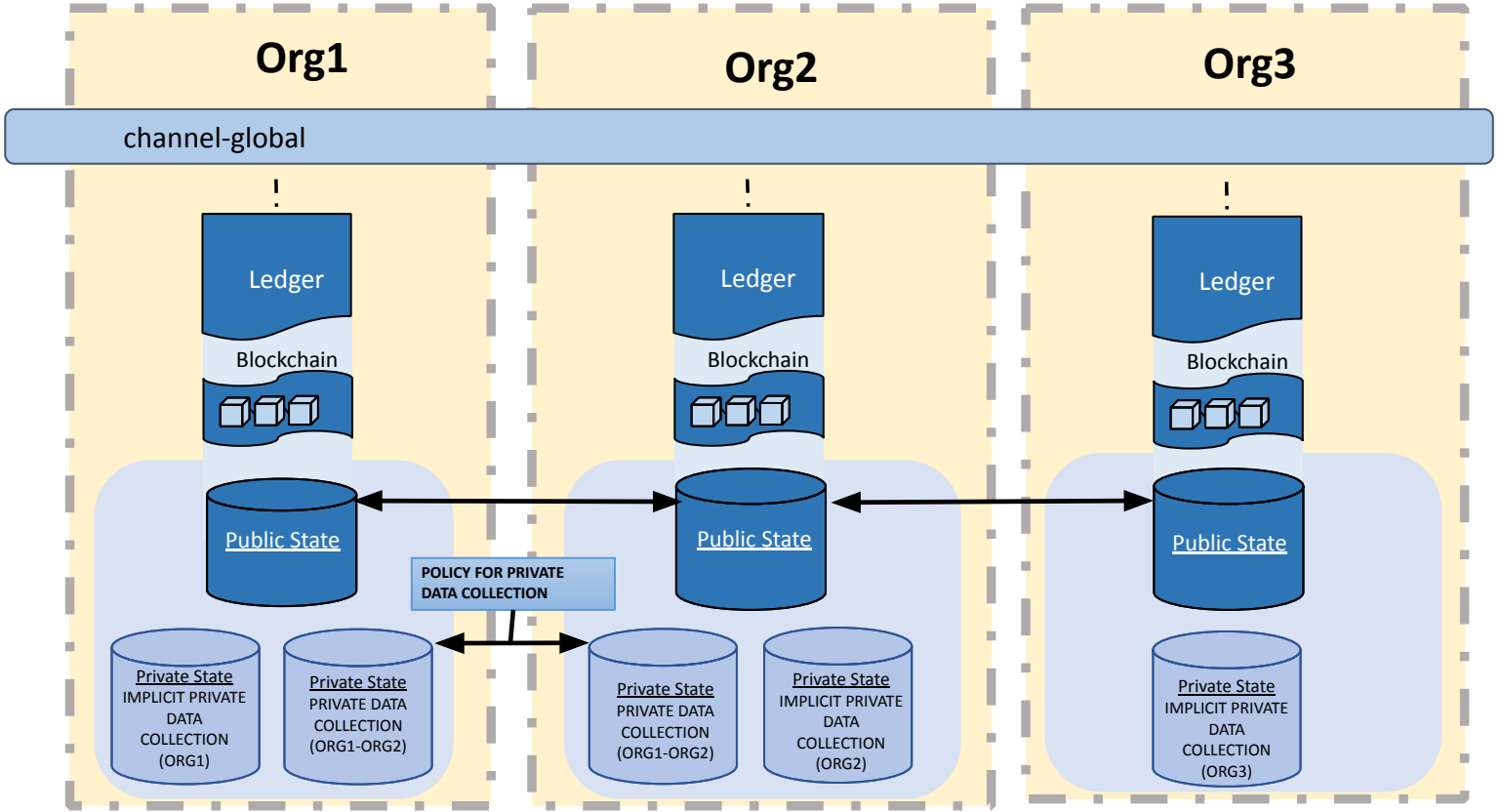


L. Olivieri, "Detection of Cross-Channel Invocation Risks in Hyperledger Fabric," *2025 IEEE 36th International Symposium on Software Reliability Engineering (ISSRE)*

Private Data Collections



Private Data Collections



Pro and Cons

- ✓ Interoperability **native intra-framework**
- ✓ **Confidentiality, privacy preservation**
- ✓ **Low system effort**

- − **Management too permissive**
  - ○ **Smart contract development → error-prone**
  - ○ **Data leakages**

Data Leaks in HF

```
args, err := stub.GetArgs()
err := stub.PutPrivateData("pdc1","keyA",args[0])
```

Public Data in Private Location!

Data Leaks in HF

```
args, err := stub.GetArgs()
err := stub.PutPrivateData("pdc1","keyA",args[0])
```

Public Data in Private Location!

```
val, err := stub.GetPrivateData("pdc1","keyA")
err := stub.PutState("keyA", val)
```

Private Data in Public Location!

. . .

Other Solutions

Weaver DLT

Weaver DLT

HYPERLEDGER CACTUS

+

Weaver DLT

=

HYPERLEDGER CACTI

General-purpose languages

- Full language features
- No restrictions

Takeaways

- Simple interoperability **transactions** and **events**.
  - may produce **unexpected behaviors**.
- Native interoperability mechanisms support **confidentiality** and **privacy**.
- Approaches can be **too restrictive or too permissive**.
- **Automated verification** is essential for correctness and security. However, there are **only few tools**.

Thanks for the Attention!

Università
Ca'Foscari
Venezia

Luca Olivieri

- Assistant Professor (no-tenure)
- Ca' Foscari University of Venice
- **Email:** luca.olivieri@unive.it

Interoperability in Enterprise Hyperledger Fabric Blockchains

8th Distributed Ledger Technology Workshop (DLT'26)

Luca Olivieri, Aradhita Mukherjee, Nabendu Chaki, Agostino Cortesi

Finanziato dall'Unione europea NextGenerationEU

Ministero dell'Università e della Ricerca

Italiadomani
PIANO NAZIONALE DI RIPRESA E RESILIENZA

SERICS
SECURITY AND RIGHTS IN THE CYBERSPACE