Microsoft

Confidential Consortium Framework (CCF)

Julien Maffre
Microsoft Research, Cambridge
Julien.Maffre@microsoft.com

09 Dec. 2020

Microsoft

CCF in a nutshell

- Open-Source framework to build confidential applications
- Replicated, using Trusted Execution Environment (Intel SGX)
- Hardware-backed integrity and confidentiality, with auditable ledger
- High availability

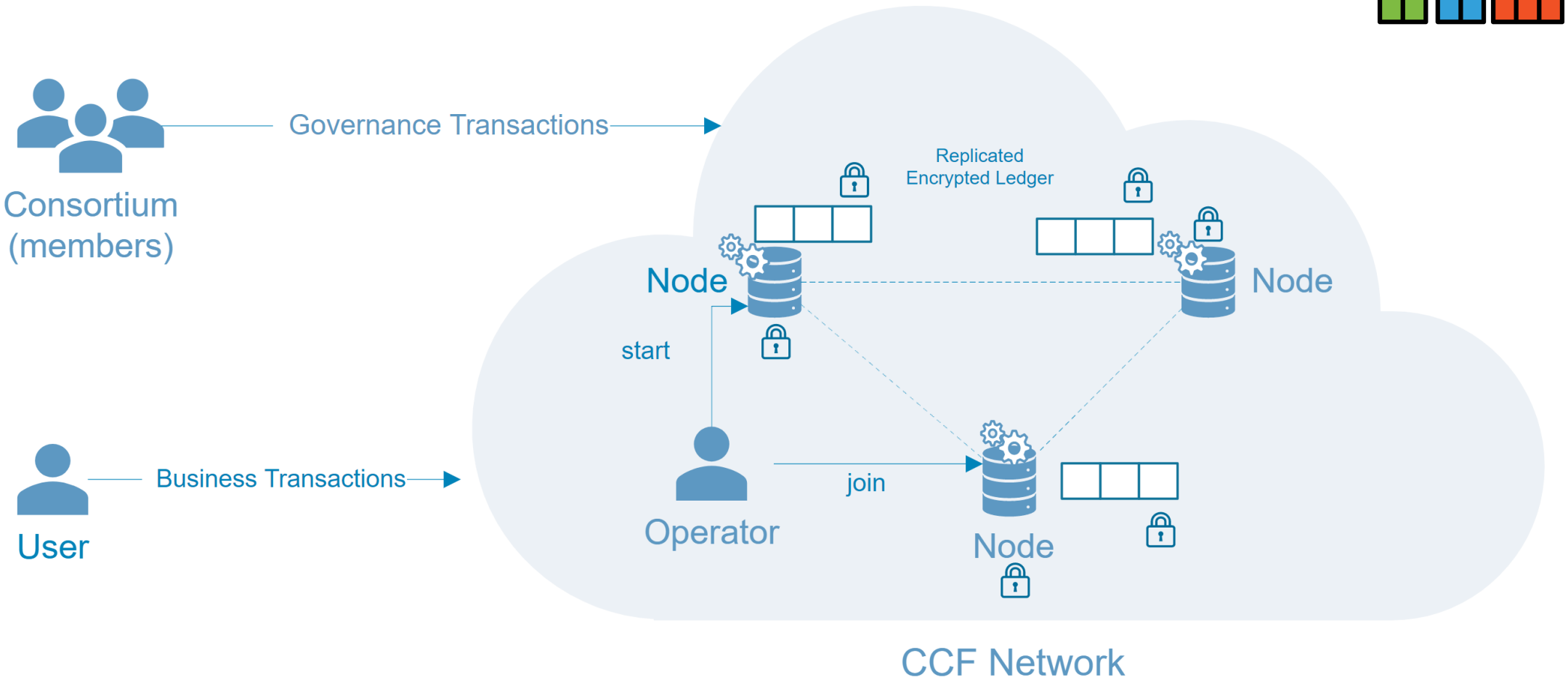
Microsoft

Motivations

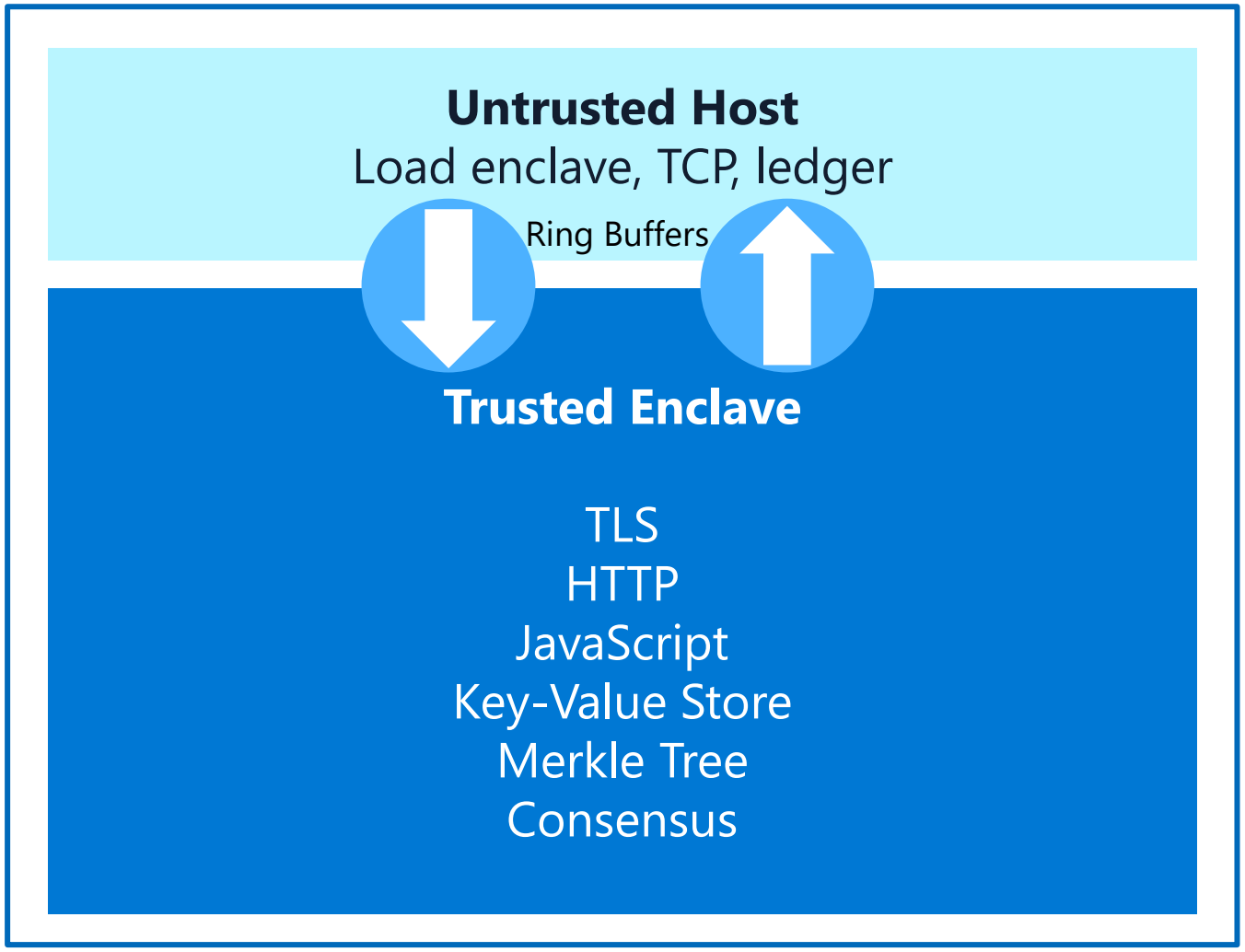
- Multi-party computation
- Confidentiality and integrity (e.g. upcoming Azure Ledger)

- Web apps: TypeScript/JavaScript, HTTPs, JWT
- Simple programming model to replicated key-value store

System Architecture

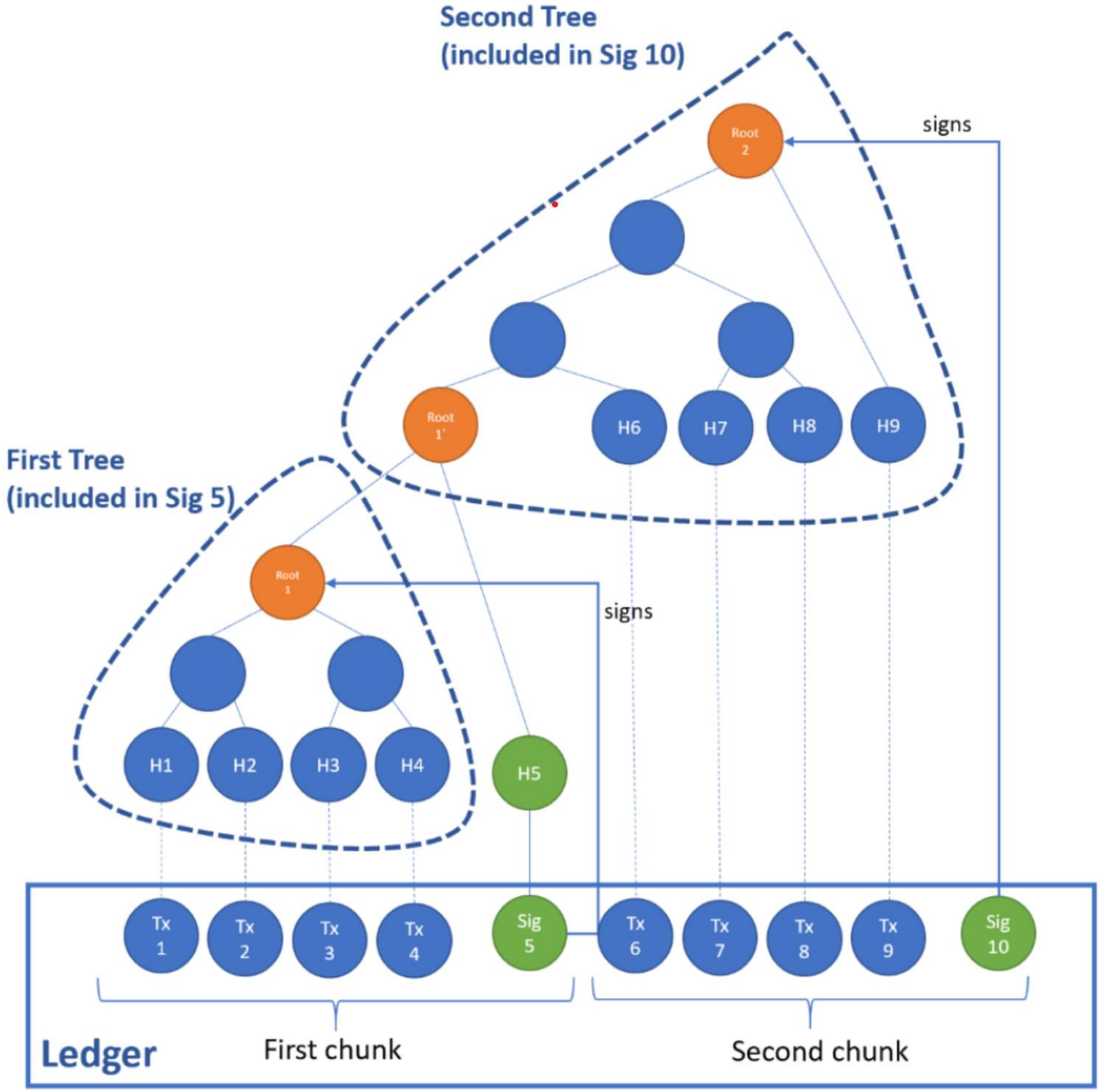


CCF Node Architecture



Verifiable Ledger

- Replicated ledger on all nodes in the network via crash or Byzantine fault tolerant consensus
- Fully auditable and verifiable offline
- Receipts



Microsoft

Member Governance

- Members ≠ Users
- Members rule the CCF service via proposals and votes
- Members vote to add new members, new code measurements, etc.
- Rules (constitution) are scriptable
- Their actions are recorded publicly for auditing

TypeScript CCF app

1. Application data (KV store)

2. HTTP endpoints

3. Business logic

```
import * as ccf from "ccf"

@Route("bank")
class MyCCFBankingApp {
  private accounts = ccf.Map("accounts", string, bigint)

  @Post("accounts")
  create_account(@Body() body: CreateAccountRequest): void {
    if (this.accounts.has(body.name)) {
      throw new Error("Account already exists!")
    }

    this.accounts.set(body.name, body.balance)
    this.setStatus(201)
  }

  @Post("transfer")
  transfer(@Body() body: TransferRequest): void {
    if (!this.accounts.has(body.src) || !this.accounts.has(body.dst)) {
      throw new Error("Src or dst account does not exist")
    }

    const src_balance = this.accounts.get(body.src)
    const dst_balance = this.accounts.get(body.dst)
    this.accounts.set(body.src, src_balance - body.value)
    this.accounts.set(body.dst, dst_balance + body.value)
    this.setStatus(200)
  }
}
```

1.

2.

3.

2.

3.

Future Work

- Byzantine identity
- Sharding
- Deployment to Azure

Microsoft

Use CCF today

GitHub
https://github.com/microsoft/CCF

Documentation
https://microsoft.github.io/CCF

Sample TypeScript app
https://github.com/microsoft/CCF/tree/master/samples/apps/forum

Open Enclave SDK
https://github.com/openenclave/openenclave

Azure Confidential Computing
https://azure.microsoft.com/en-gb/solutions/confidential-compute/

Microsoft

Thanks

Questions welcome!