E-Voting Challenges

15.12.2025

Céline Camacho
E-Voting Software Engineer

Nils Aellen
E-Voting Solution Architect

Swiss Post

- **Physical and digital world**
- **Bound to mail secrecy & trusted**
- **Cryptography competence center**

E-Voting

- **Codebase bought in 2020**
- **Productive again since June 2023**

E-Voting Phases

Setup

Configure election

Generate keys

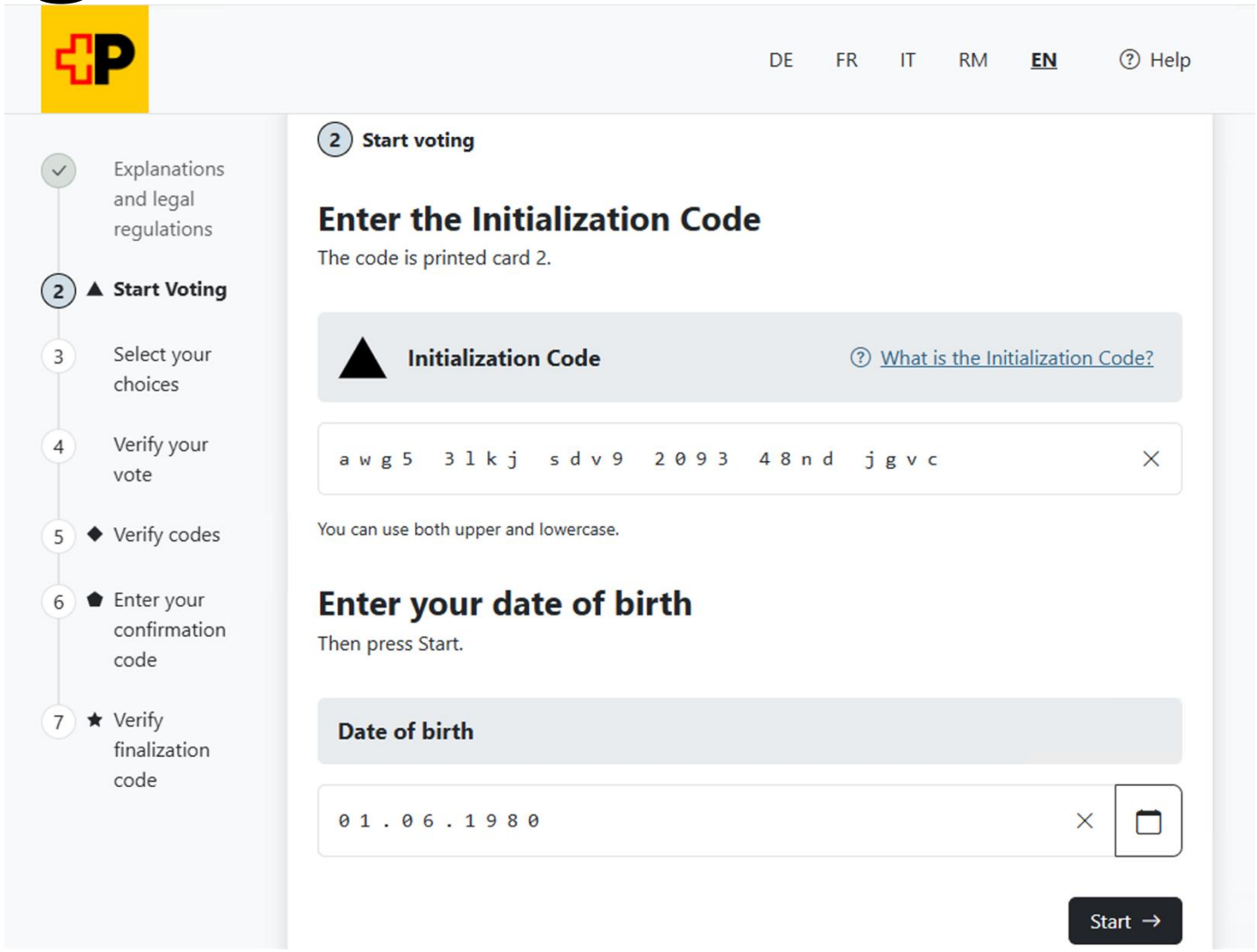
Generate return codes

E-Voting Phases

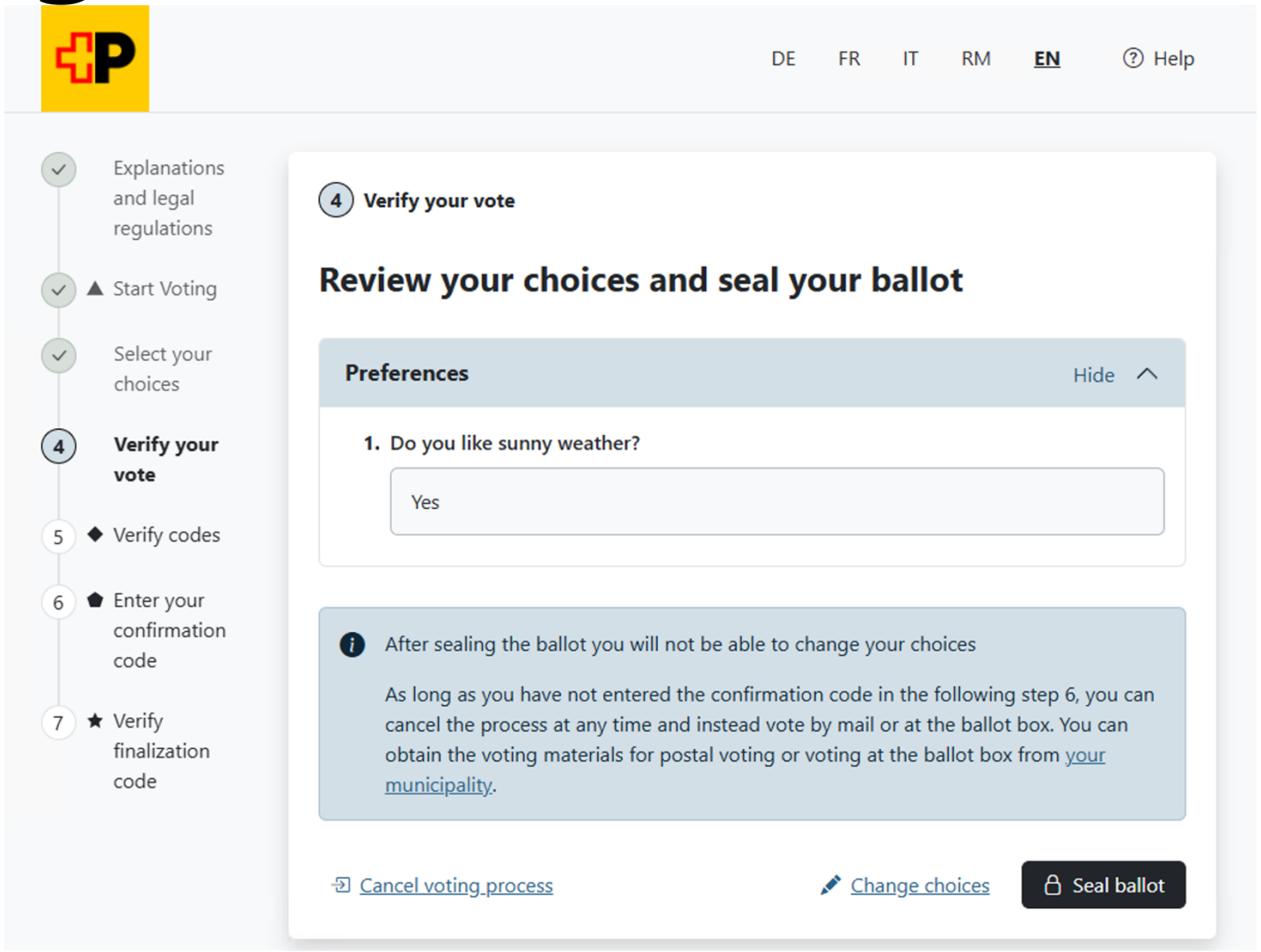
Voting

1. Authenticate
2. Send vote
3. Confirm vote

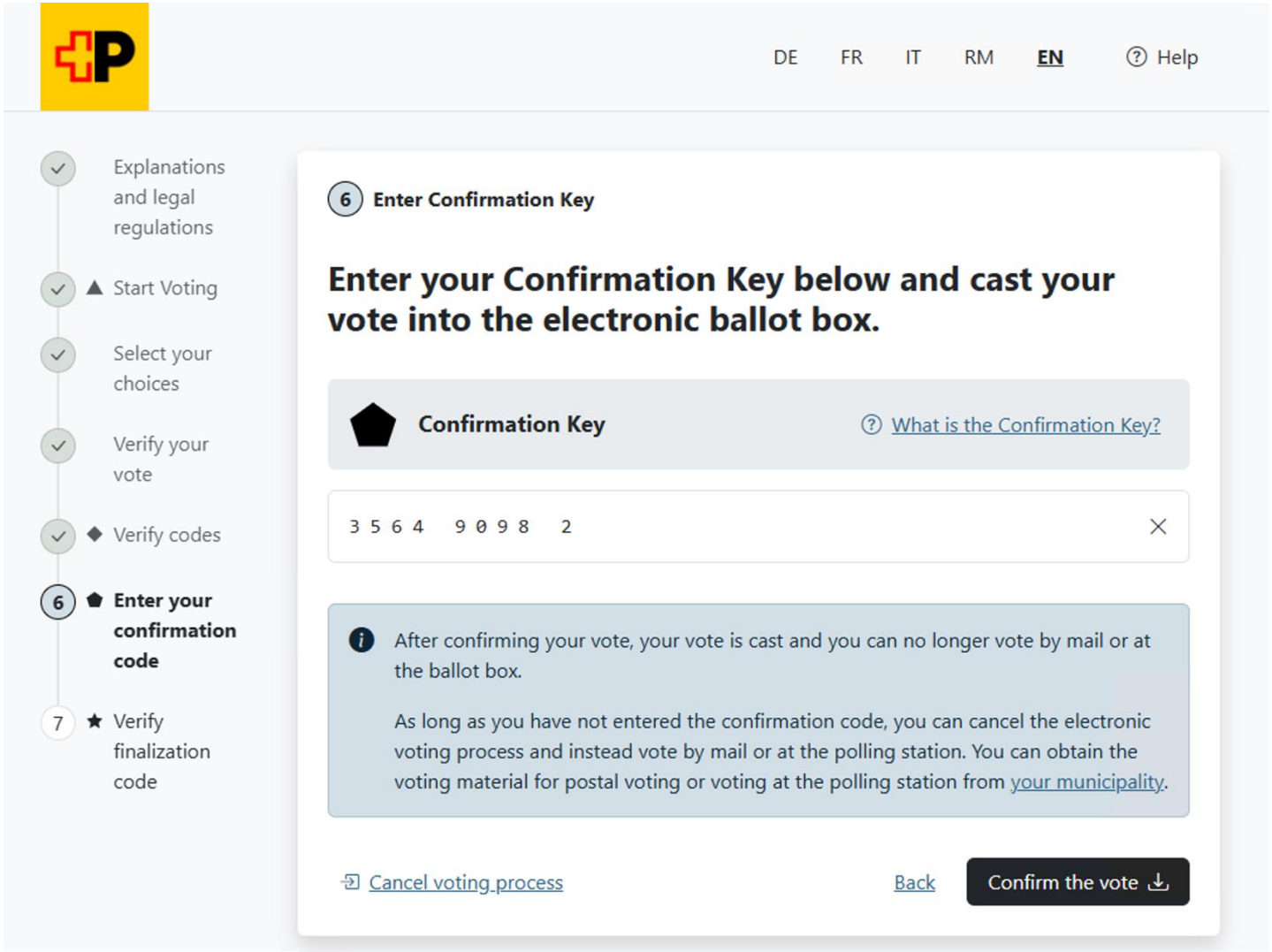
E-Voting Phases



E-Voting Phases



E-Voting Phases



E-Voting Phases

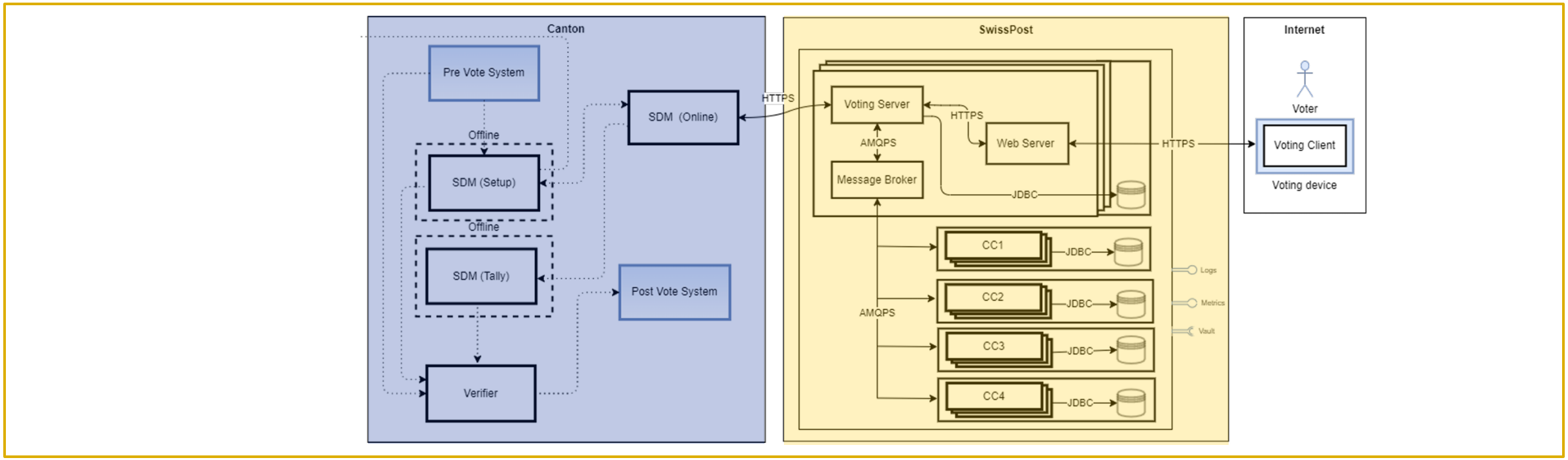
Tally

Mix

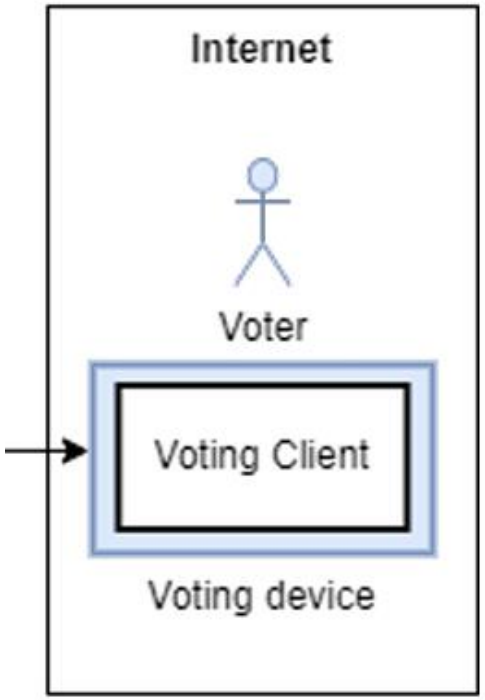
Decrypt

Results of election

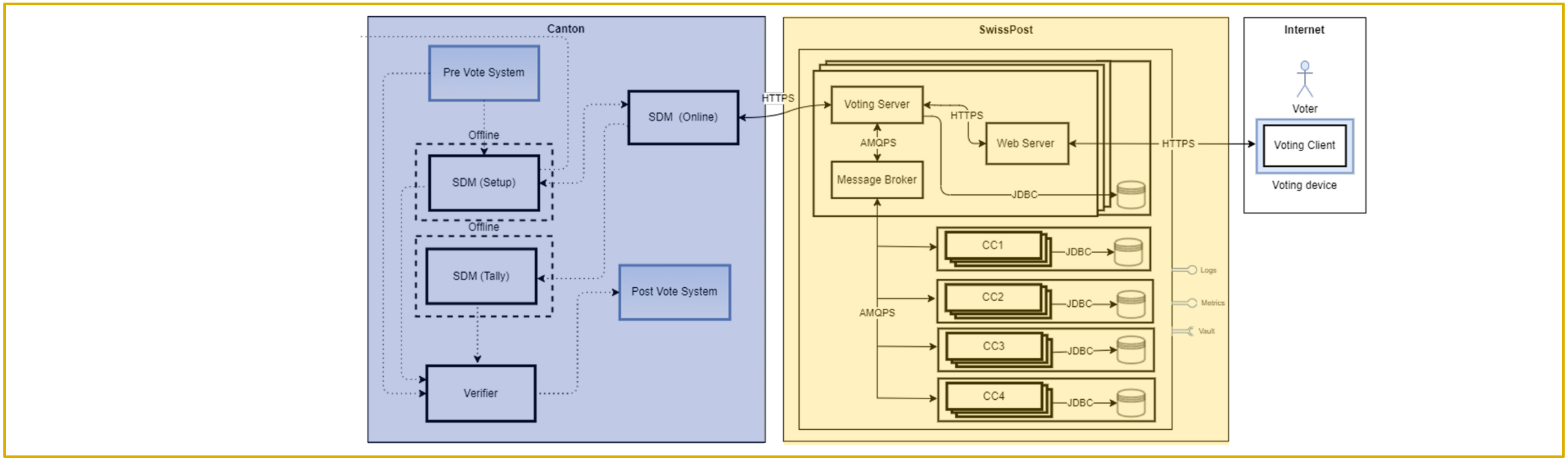
Swiss Post E-Voting



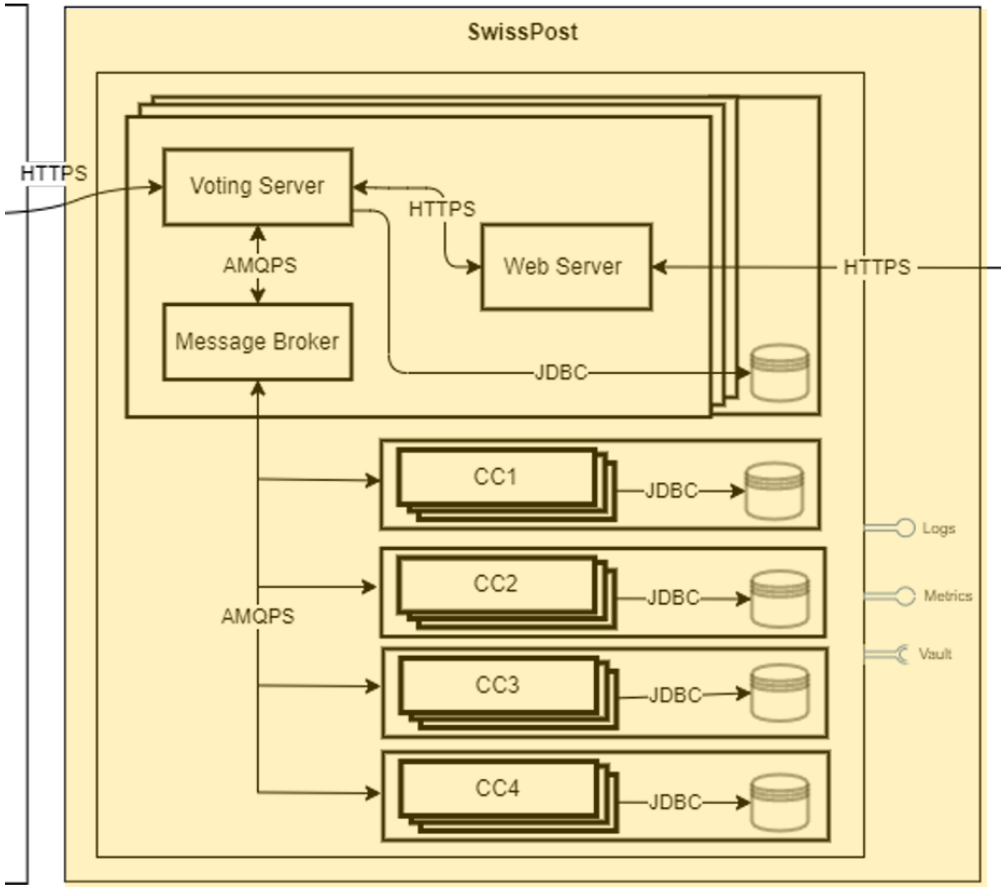
Swiss Post E-Voting



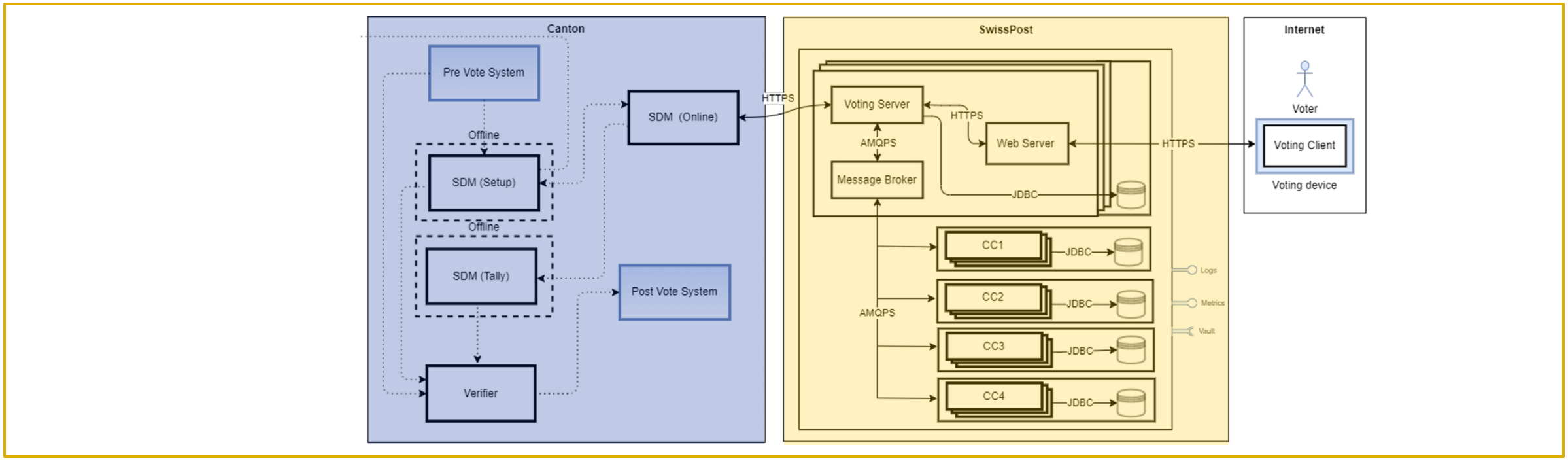
Swiss Post E-Voting



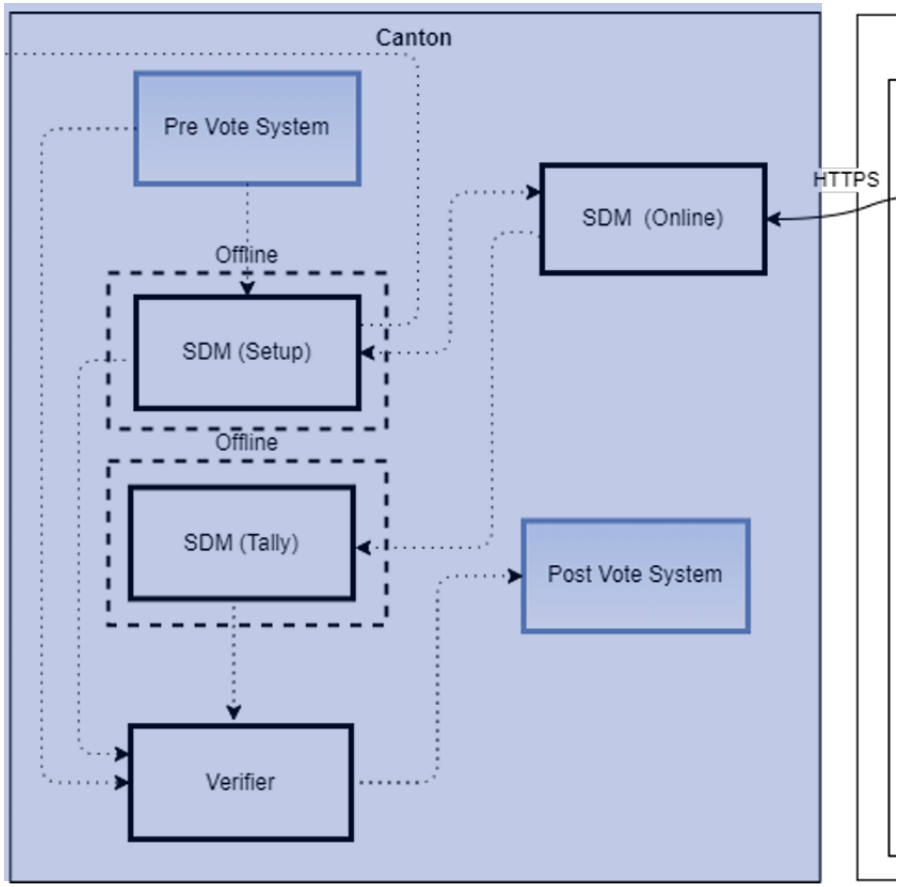
Swiss Post E-Voting



Swiss Post E-Voting

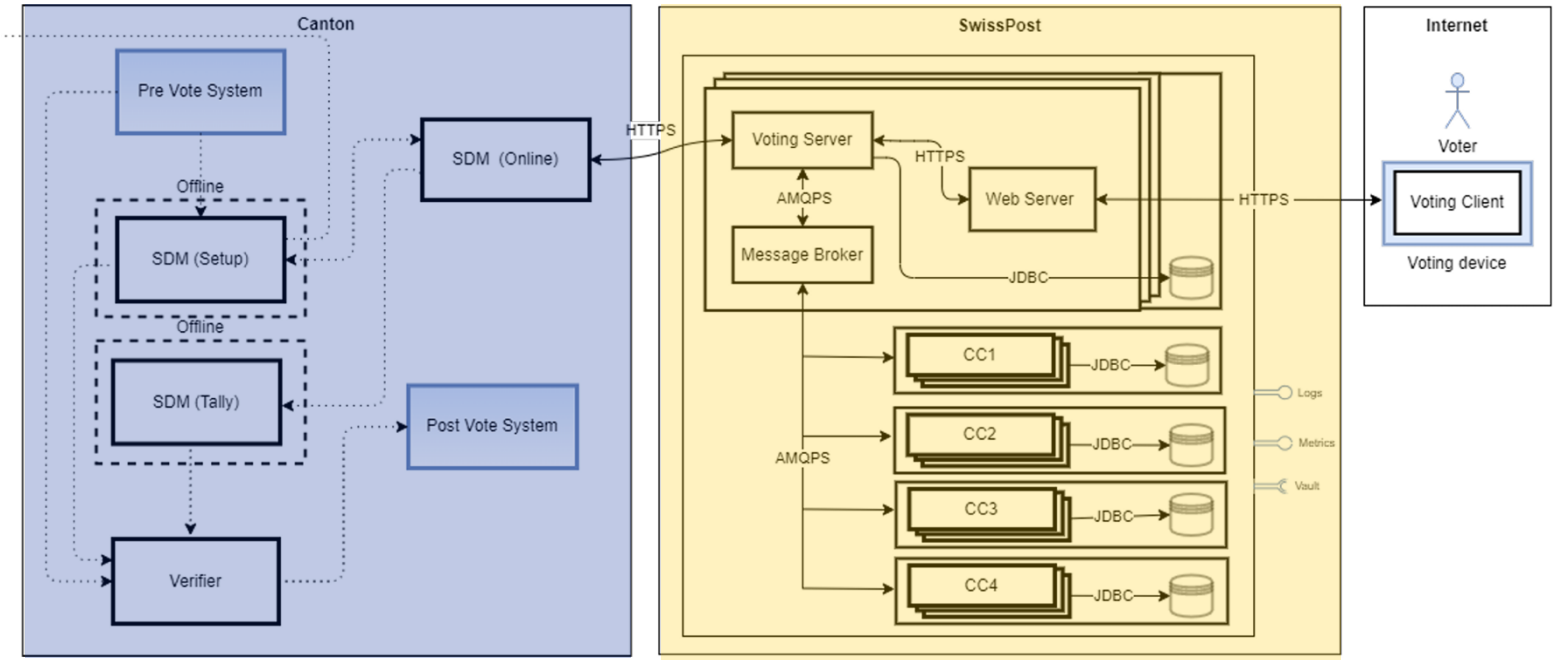


Swiss Post E-Voting



Challenges

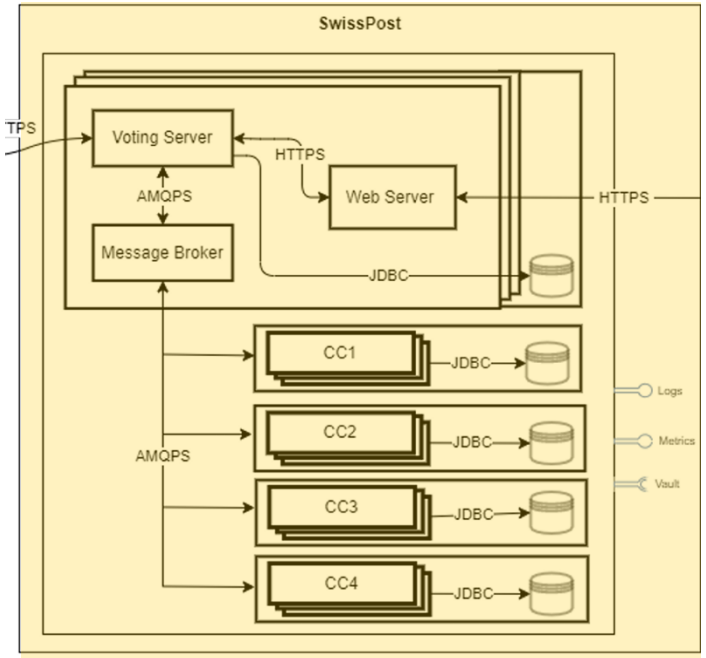
1. **Scalability**
2. **Data transfer**
3. **Resiliency**



1 Challenge – Scalability

From Monolith to Multi-Tenant SaaS

Canton 1

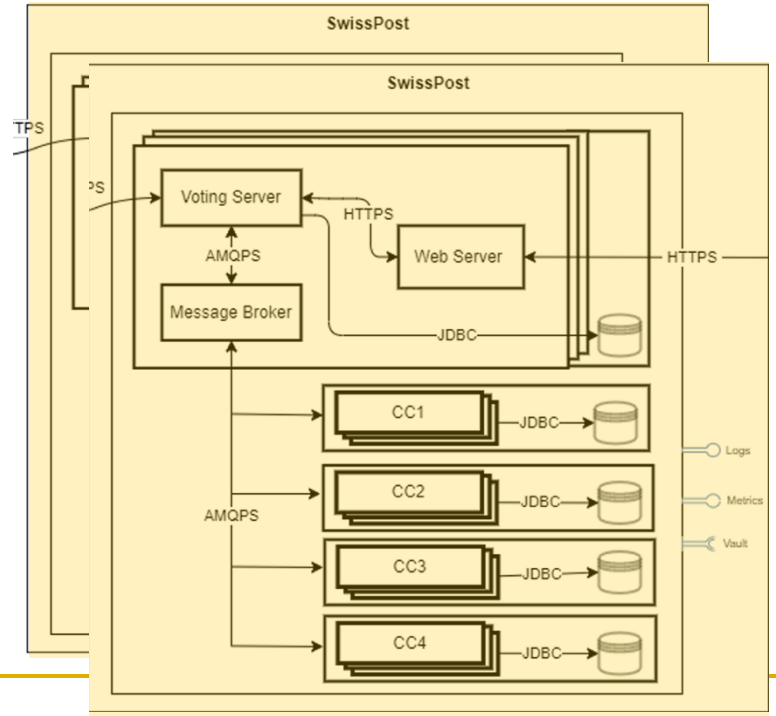


1 Challenge – Scalability

From Monolith to Multi-Tenant SaaS

Canton 1

Canton 2



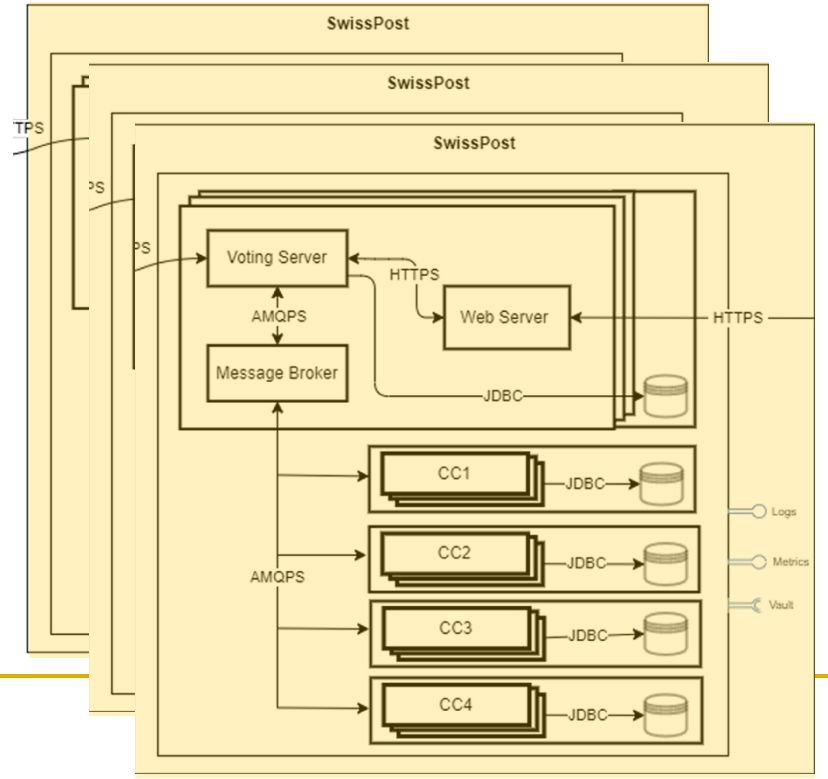
1 Challenge – Scalability

From Monolith to Multi-Tenant SaaS

Canton 1

Canton 2

Canton 3



1 Challenge – Scalability

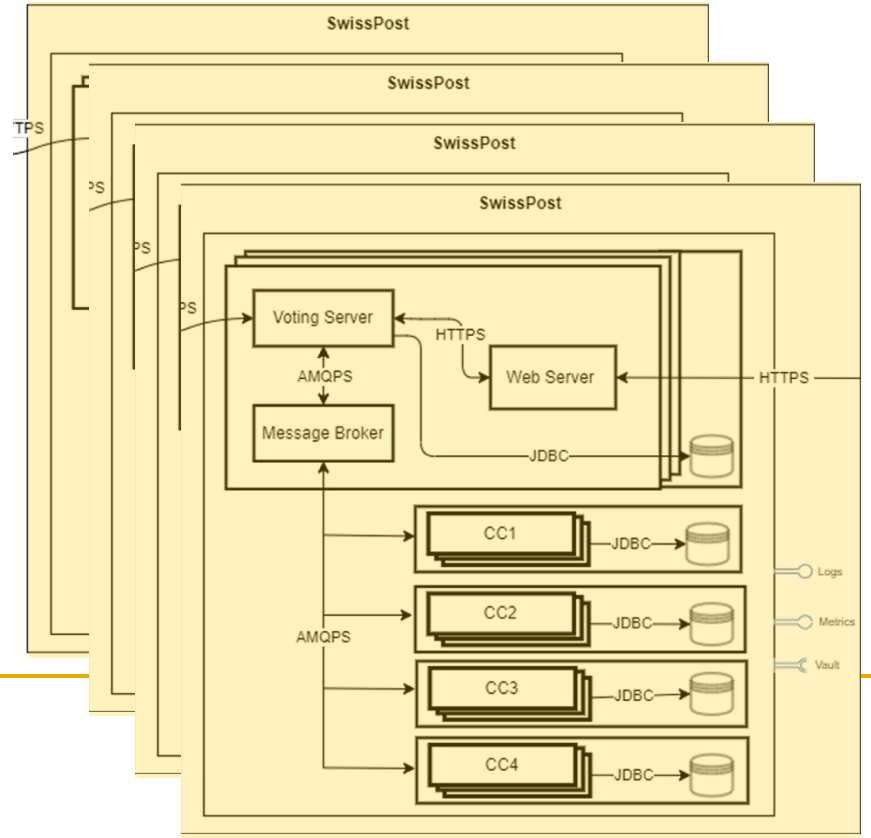
From Monolith to Multi-Tenant SaaS

Canton 1

Canton 2

Canton 3

Canton 4



1 Challenge – Scalability

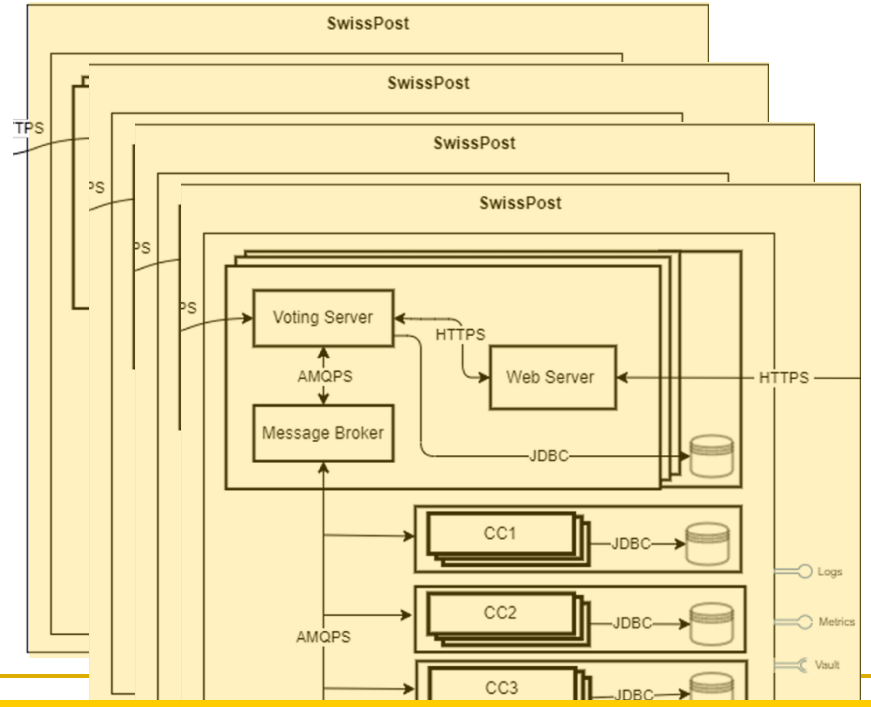
From Monolith to Multi-Tenant SaaS

Canton 1

Canton 2

Canton 3

Canton 4



More customers means more infrastructure
→ Scalable Architecture for Modern Applications

The 4 Levels of SaaS Maturity

Level 1

- 26 different solutions
- 26 different infrastructures

→ Total duplication

The 4 Levels of SaaS Maturity

Level 2

- 1 configurable solution
- 26 different infrastructures

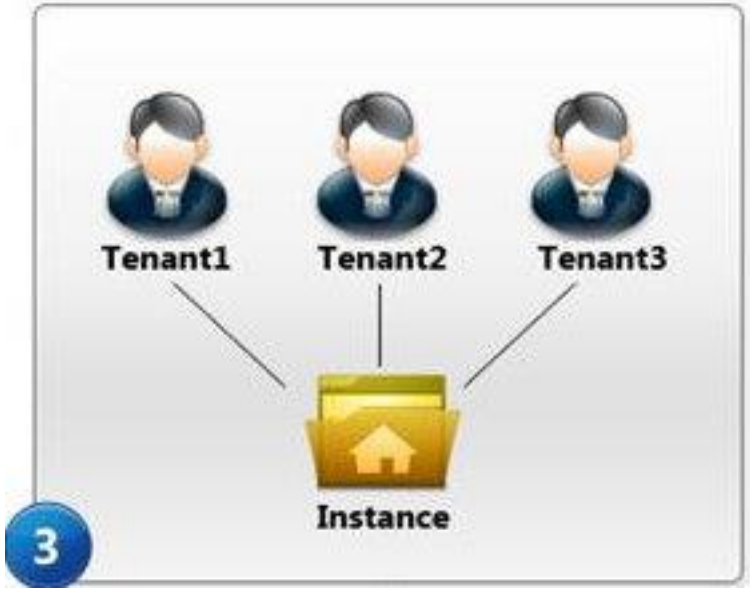
→ Unified source code

The 4 Levels of SaaS Maturity

Level 3

- 1 configurable solution
- 1 shared infrastructure

→ Multi-tenancy



The 4 Levels of SaaS Maturity

Level 4

- 1 configurable solution
- 1 auto-scalable infrastructure

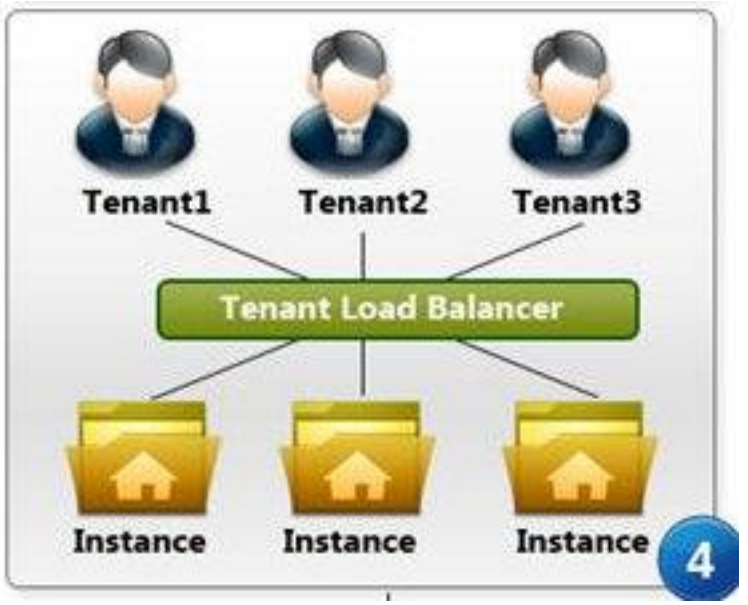
→ Mature SaaS

The 4 Levels of SaaS Maturity

Level 4

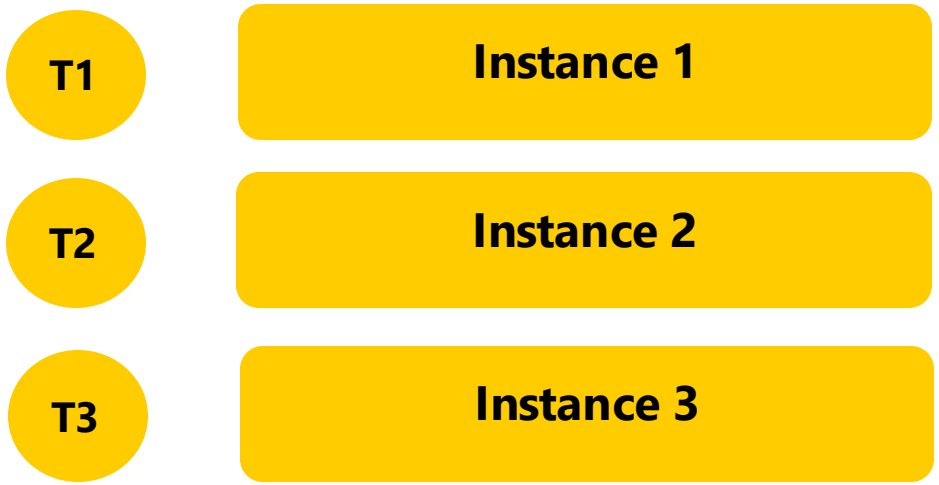
- 1 configurable solution
- 1 auto-scalable infrastructure

→ Mature SaaS



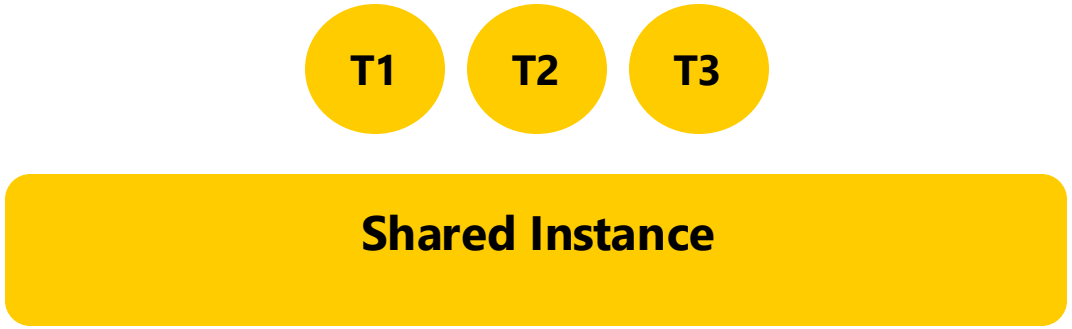
Single-Tenant vs Multi-Tenant

Single-Tenant



- ✅ Complete isolation
- ❌ Major costs
- ❌ Complex maintenance

Multi-Tenant



- ✅ Economies of scale
- ✅ Simplified maintenance
- ❌ Isolation to manage

Key Technical Aspects

Tenant identification

- Claim JWT
- URL Parameter
- HTTP Header
- Sub-domain

Database strategies

Shared tables

tenant_id column present in each table

A database per tenant

Complete data isolation

1 Key Takeaways

🎯 Objectives

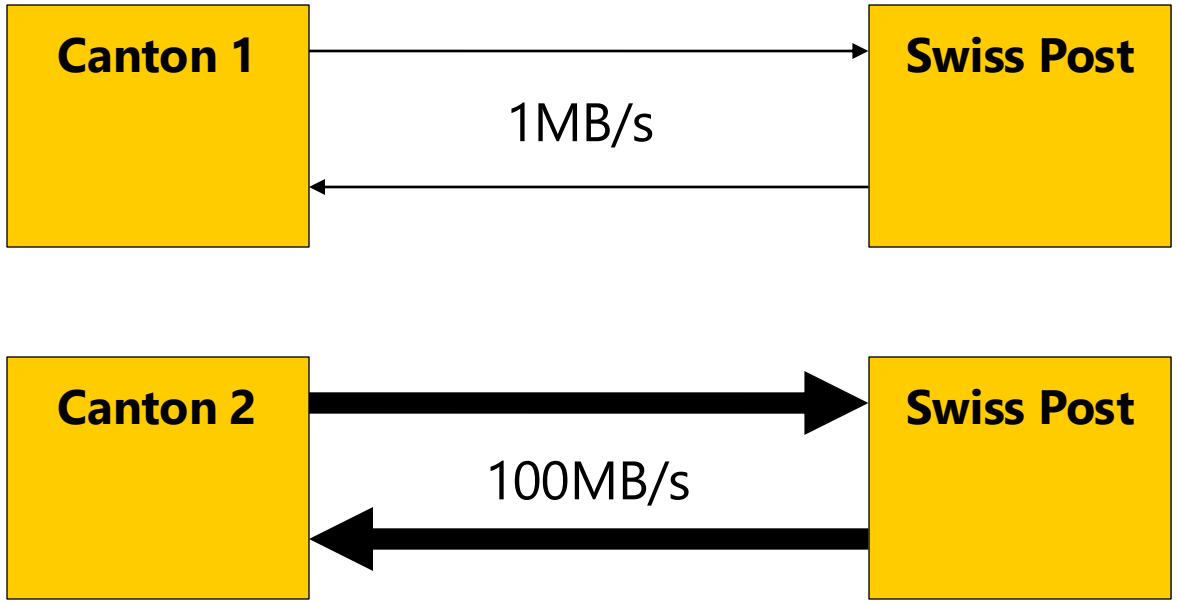
- Costs reduction
- Simplify maintenance
- Enable scaling
- Standardize code

⚖️ Trade-offs

- Increased complexity
- Security to strengthen
- Data isolation
- Per-tenant configuration

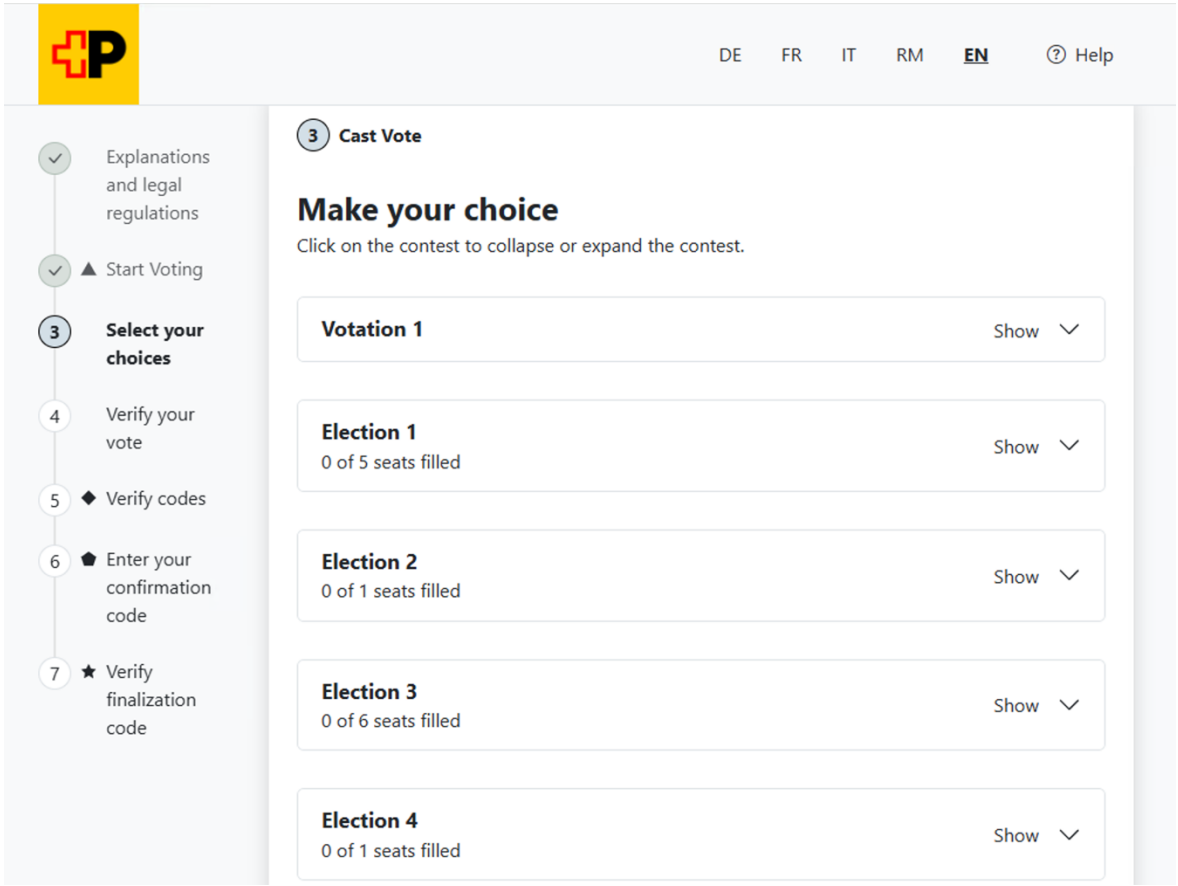
Multi-tenancy is a major architectural choice that must be planned from the design phase

2 Challenge – Data transfer



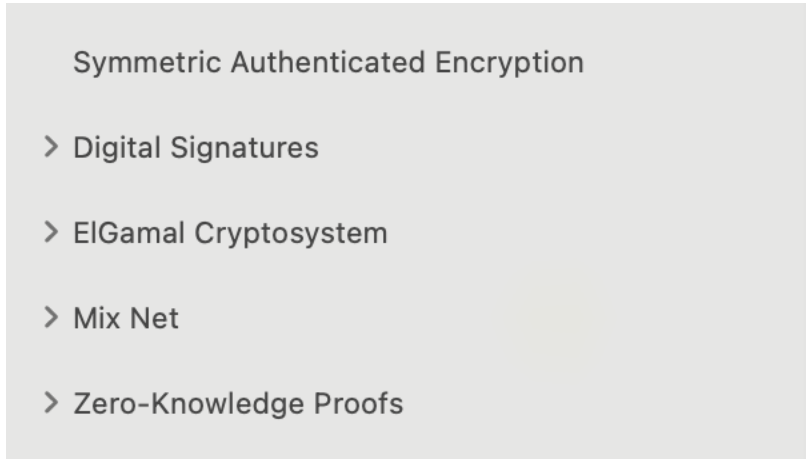
- Bandwidth depends on the worst infrastructure

2 Challenge – Data transfer



- Bandwidth depends on the worst infrastructure
- Data depends on size and complexity of election

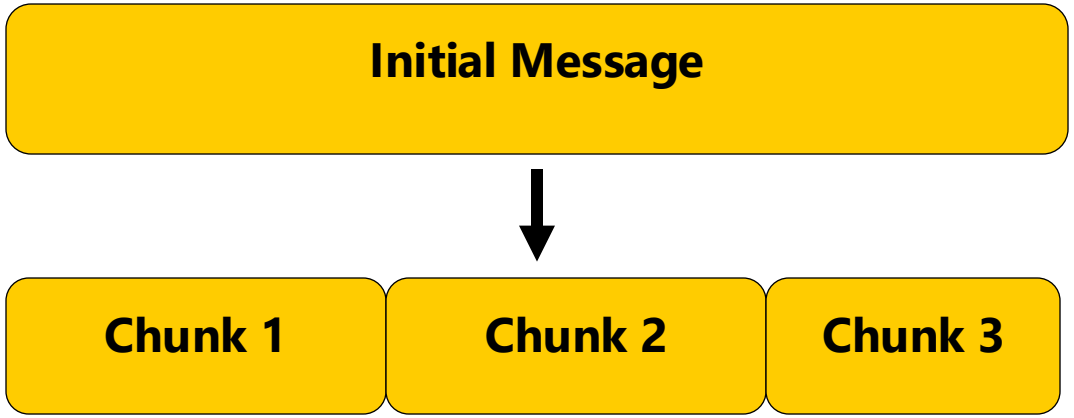
2 Challenge – Data transfer



- Bandwidth depends on the worst infrastructure
- Data depends on size and complexity of election
- Certified and recognized cryptography generates voluminous data

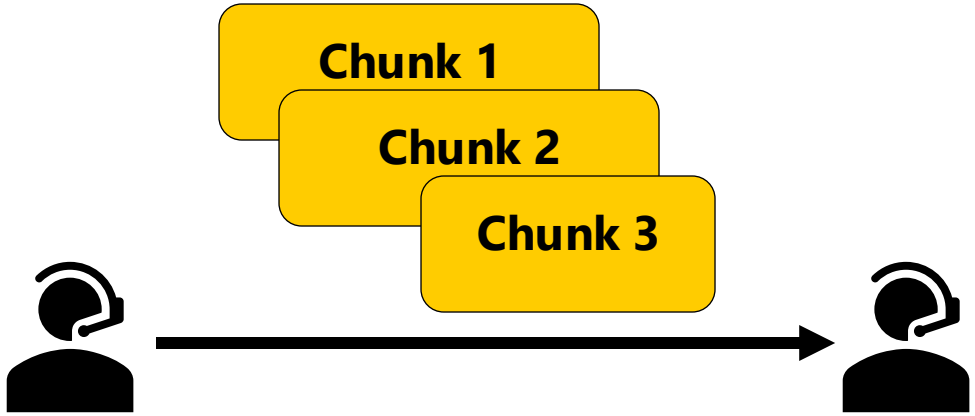
Chunking

Break large messages into smaller parts



- ✅ Avoid timeouts
- ✅ Reduce memory load
- ✅ Enables streaming

Sequential or Parallel

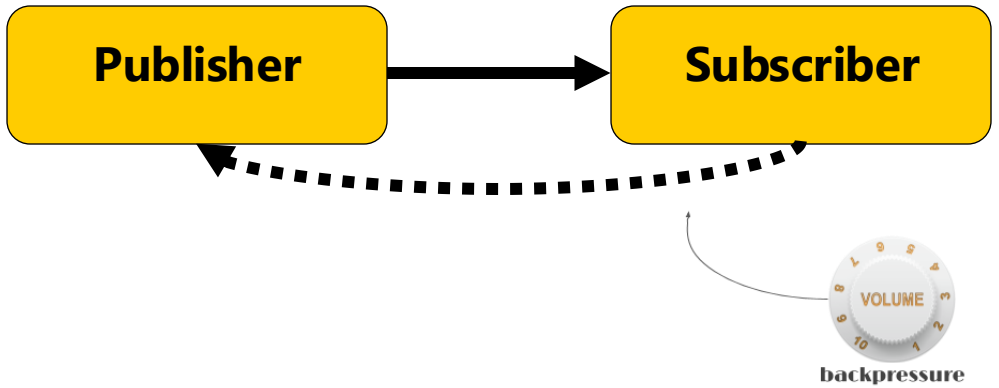


Reactive Streams & Backpressure

- Asynchronous data flow
- Backpressure signals when receiver is overwhelmed
- System adapts dynamically to load

✅ Prevents overload
✅ Improves resilience
✅ Enables real-time responsiveness

Backpressure



Key Technical Aspects

Chunk Parameters

Chunk size

Many small chunks vs. Fewer big chunks

Chunk order

Needed to reconstruct messages

Backpressure Mechanisms

Request

Receiver requests messages

Limit

Receiver limits number of messages

2 Key Takeaways

🎯 Objectives

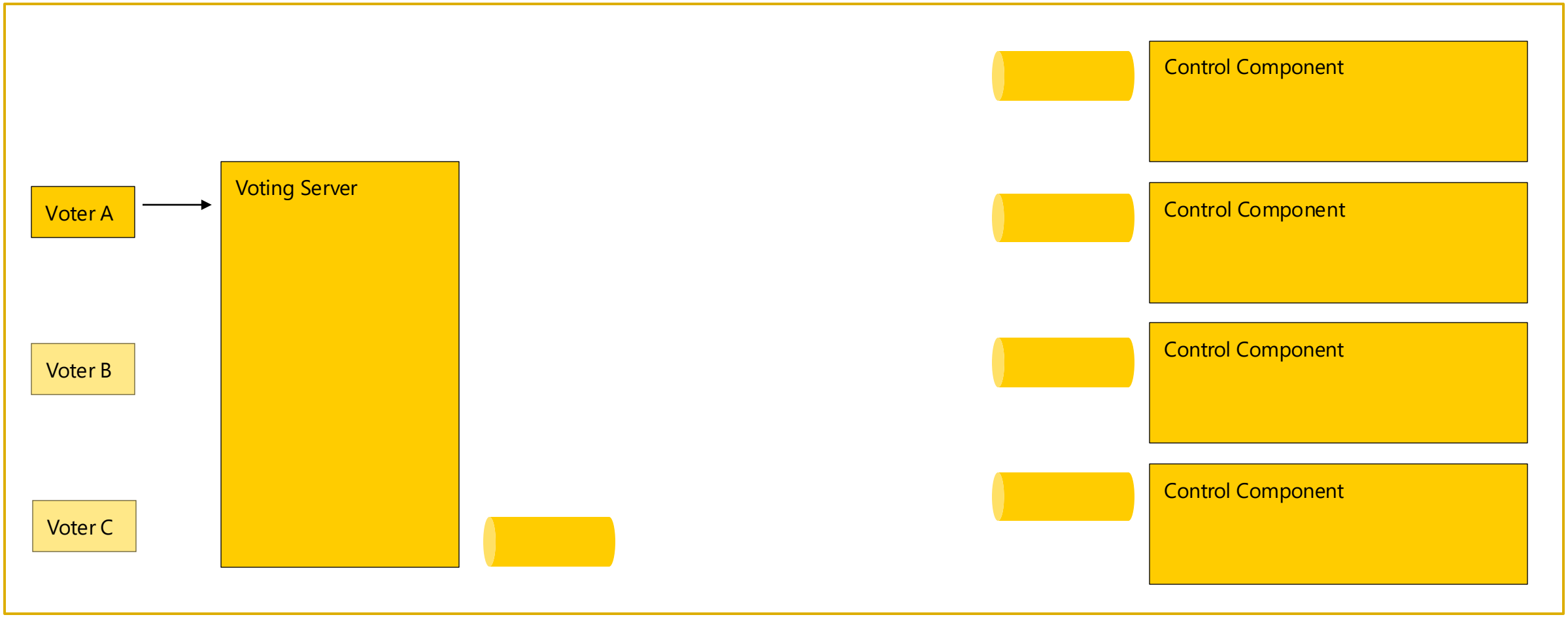
- Stable communication
- Scalability
- Improve performance
- Prevent crashes

⚖️ Trade-offs

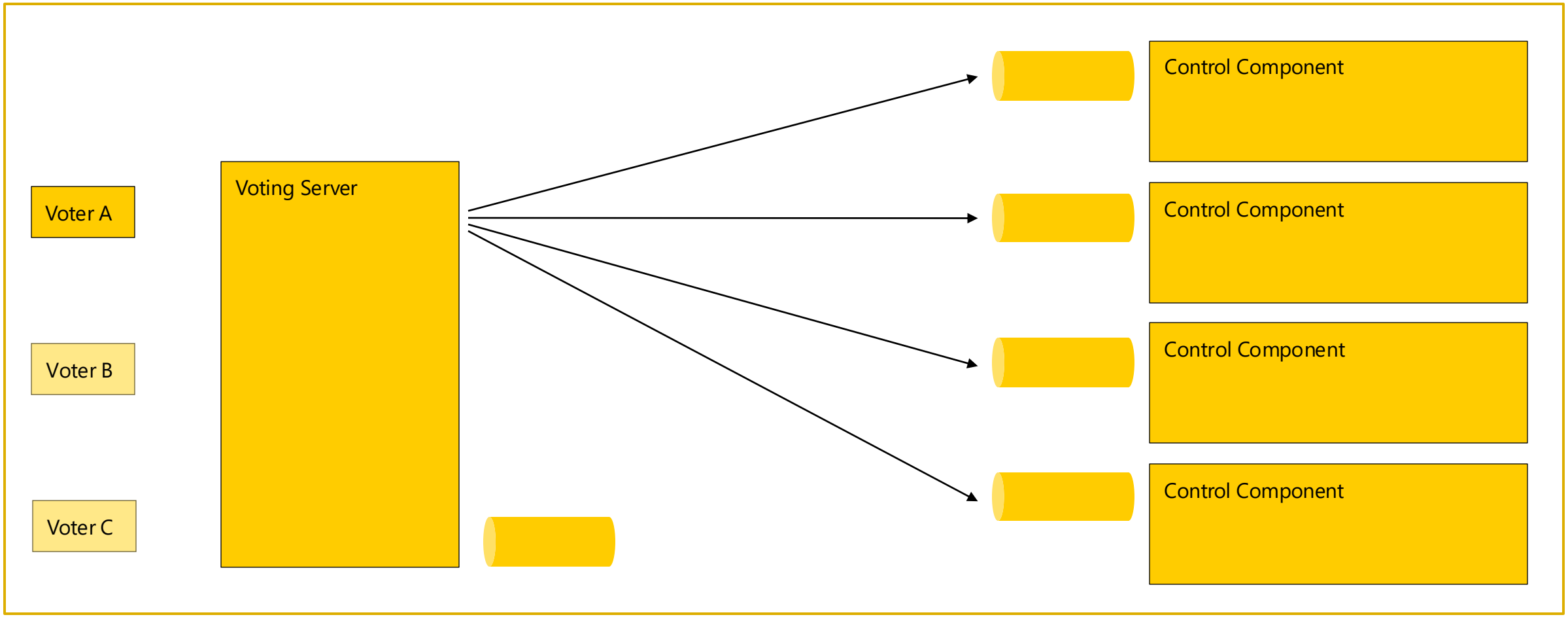
- Increased complexity
- Debuggability
- Error handling

The amount of data transmitted is a design consideration when building a scalable and resilient communication layer

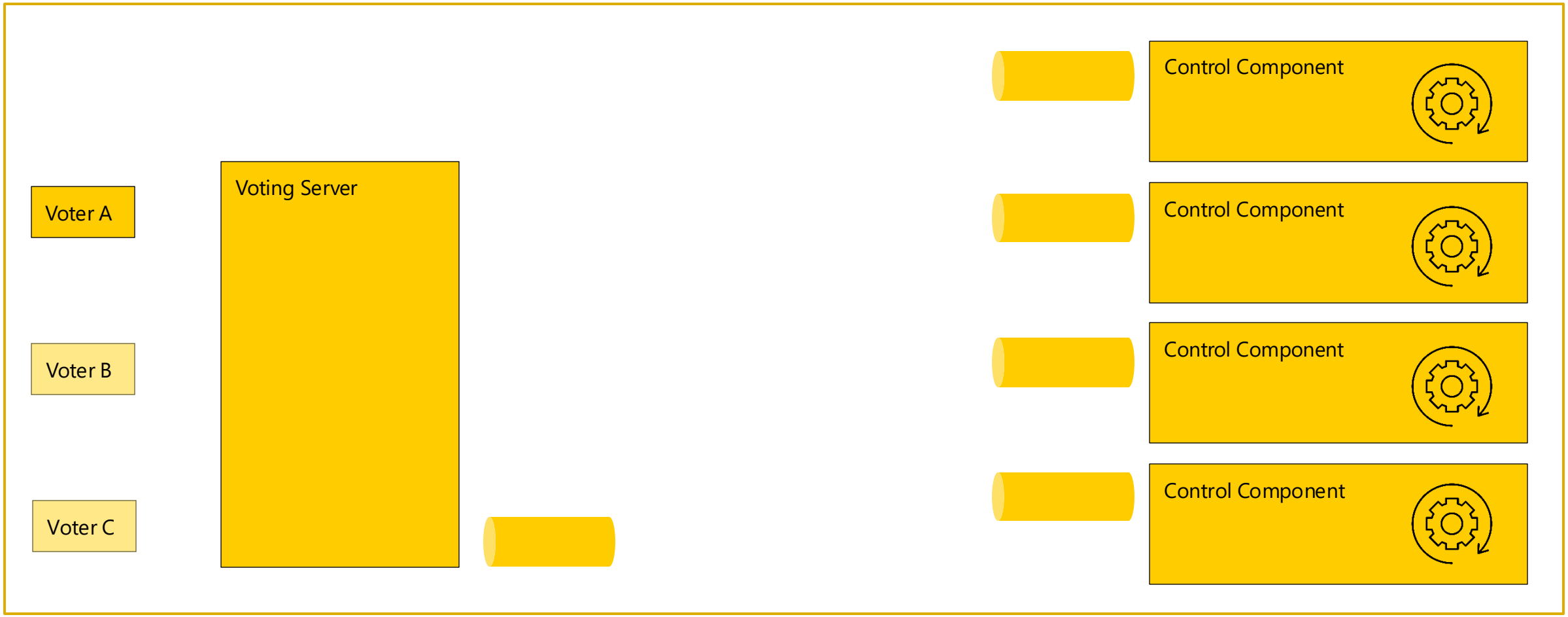
3 Challenge – Resiliency



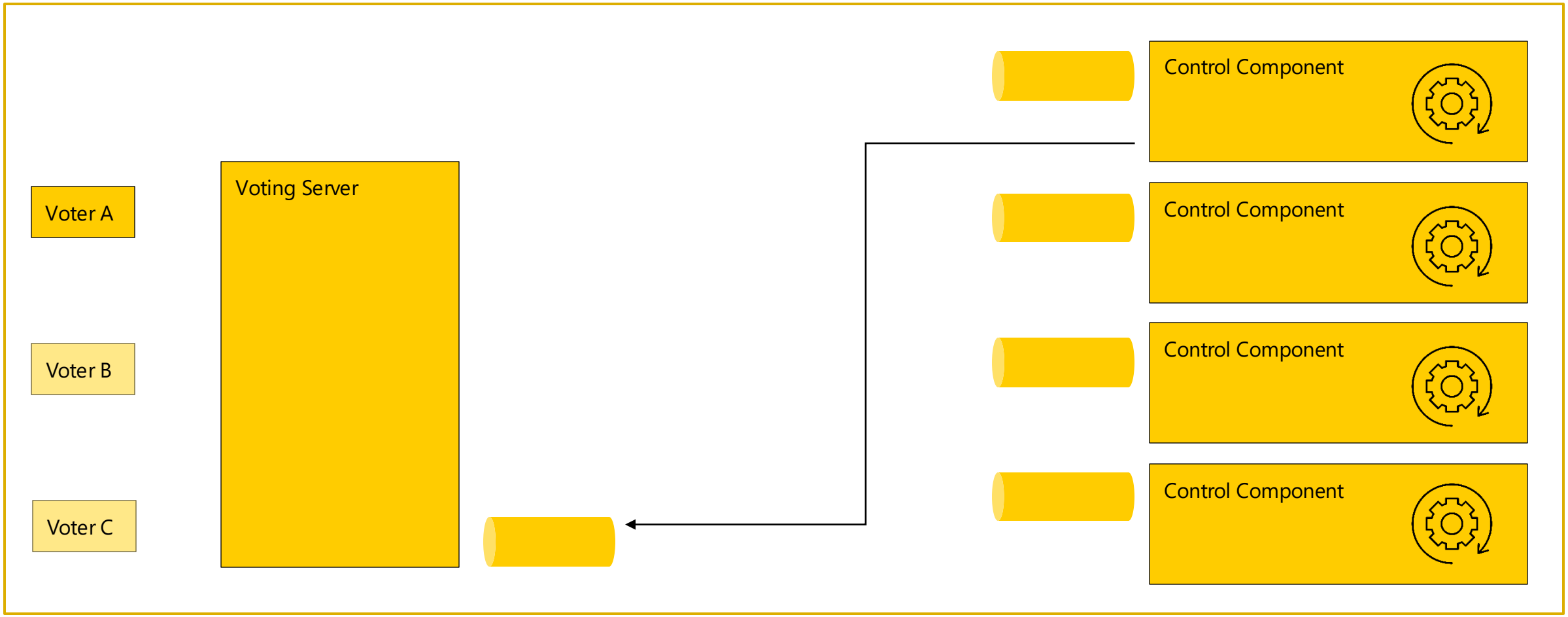
3 Challenge – Resiliency



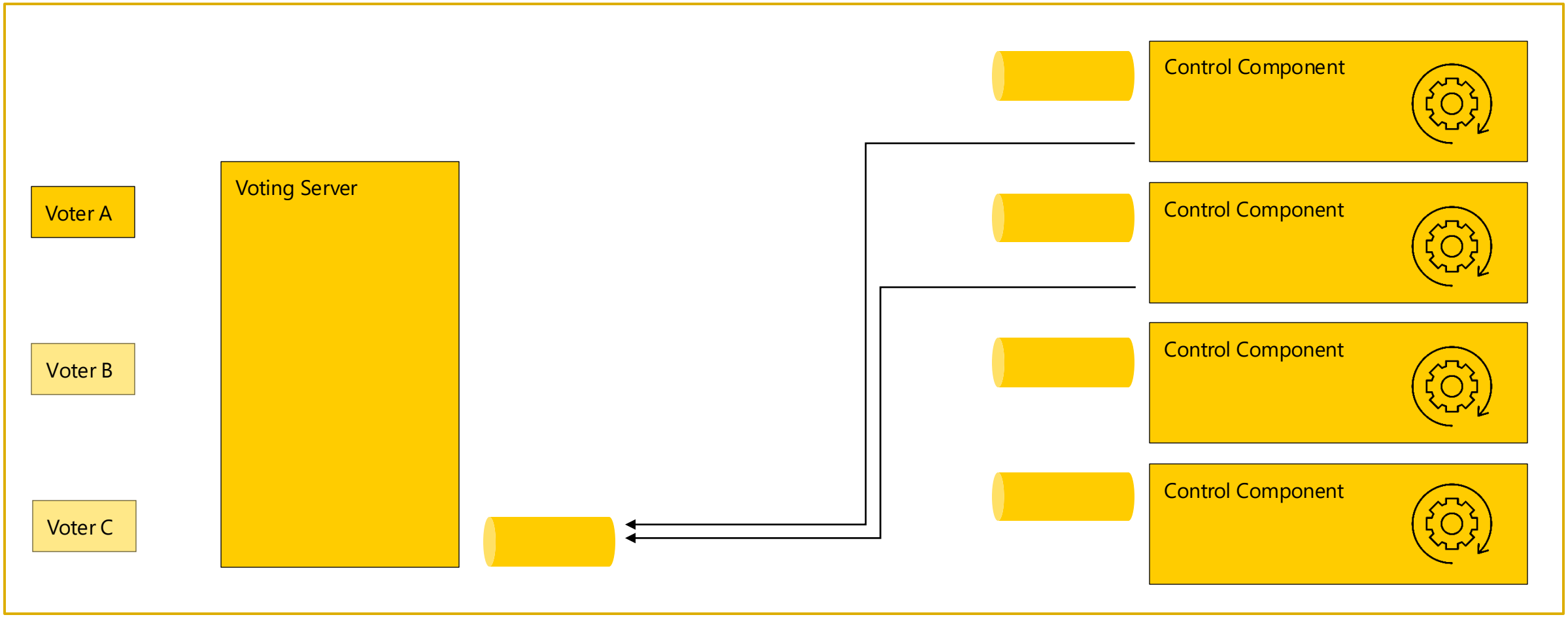
3 Challenge – Resiliency



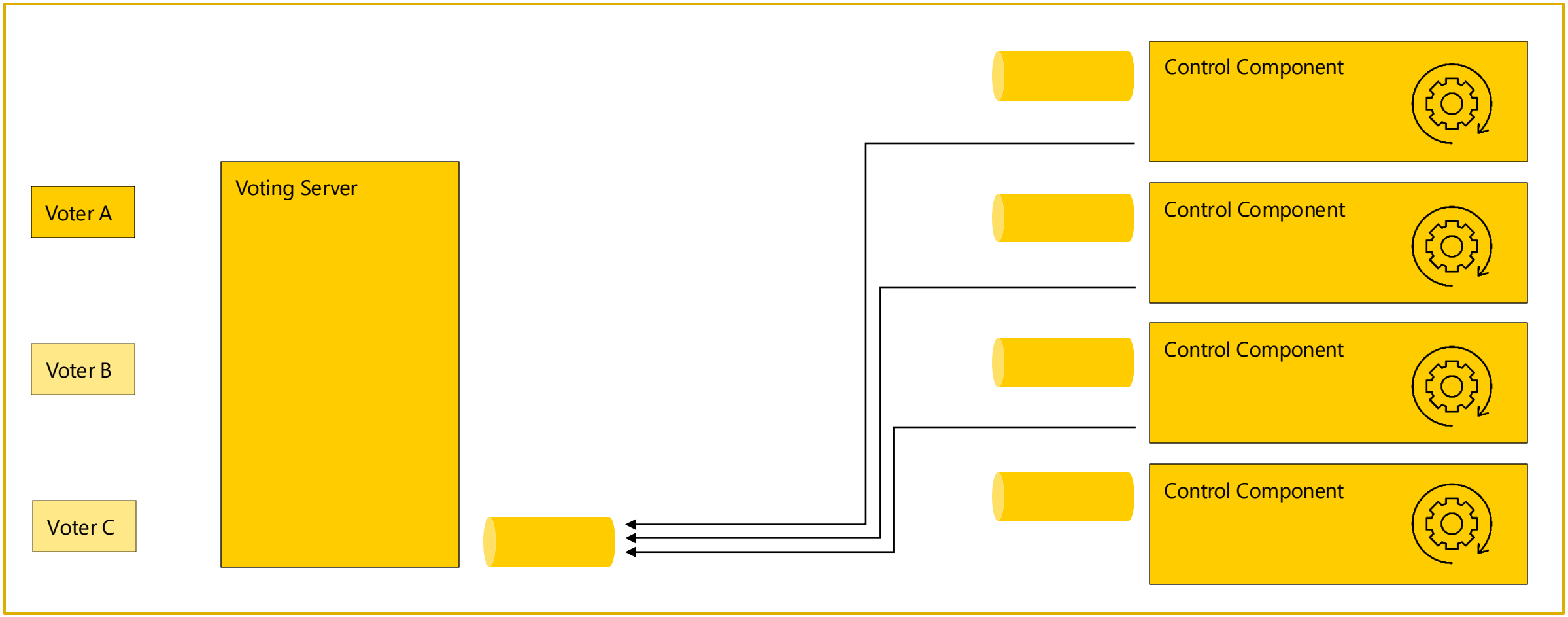
3 Challenge – Resiliency



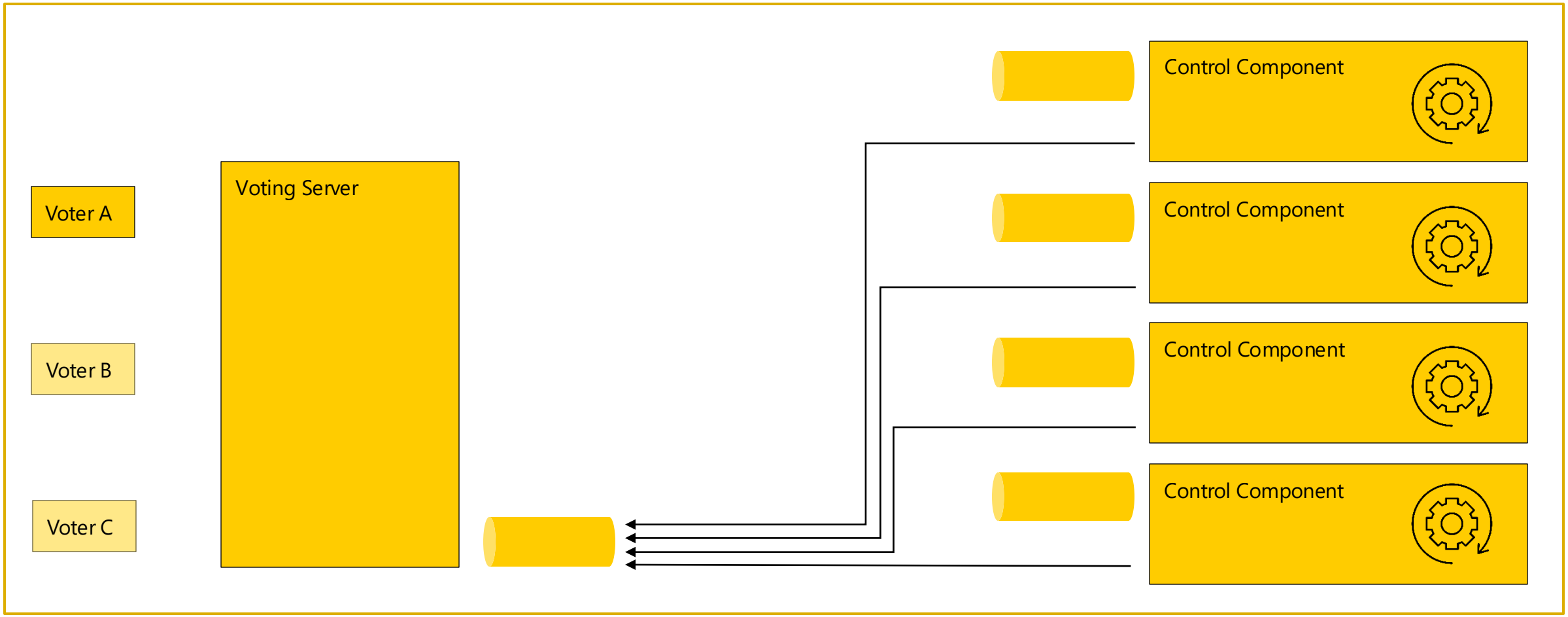
3 Challenge – Resiliency



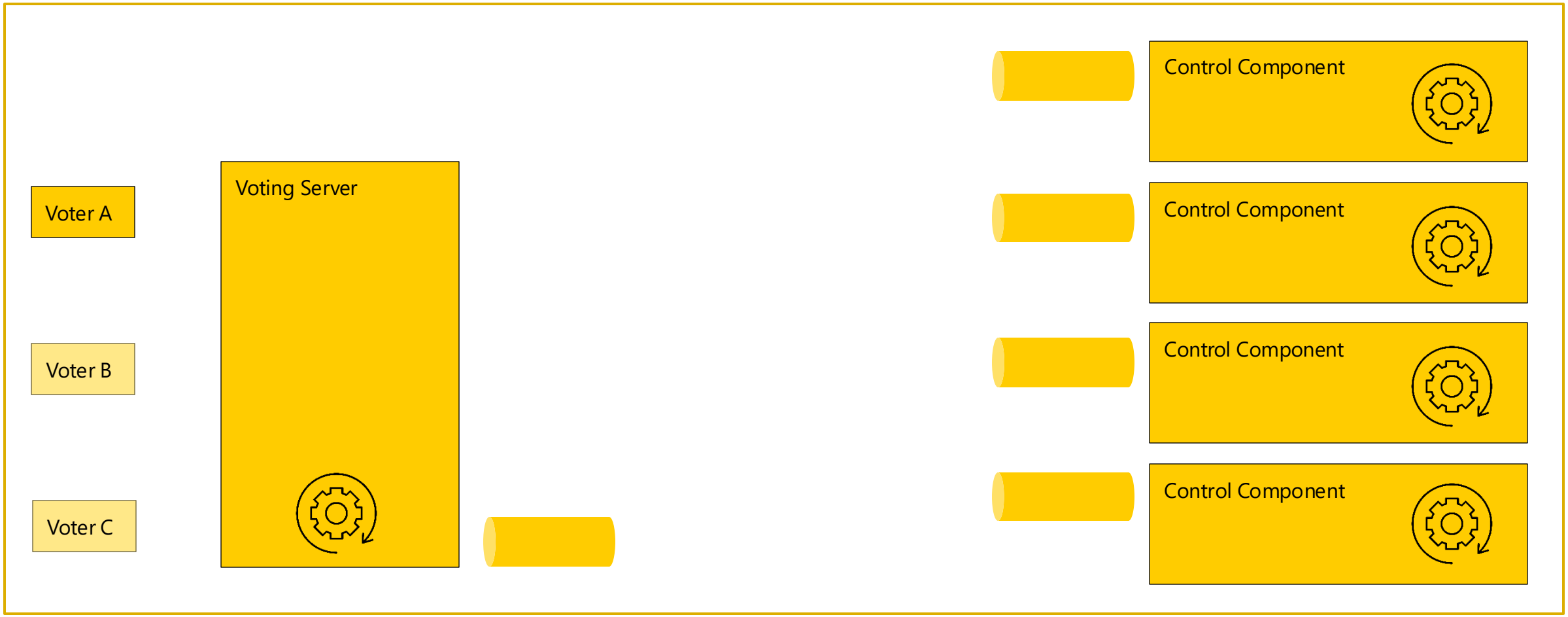
3 Challenge – Resiliency



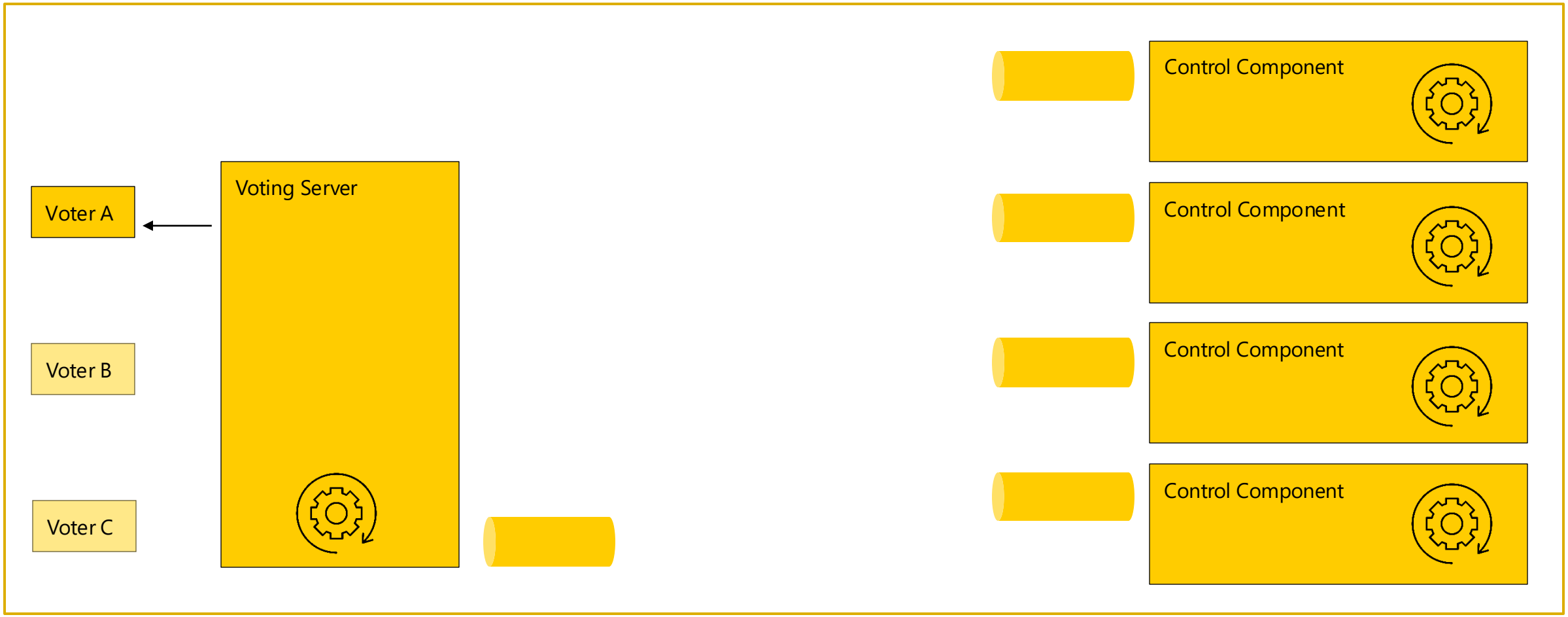
3 Challenge – Resiliency



3 Challenge – Resiliency



3 Challenge – Resiliency



Voting Server

Control Component

Control Component

send vote

Voting Server

Control Component

Control Component

send vote

Voting Server

Control Component

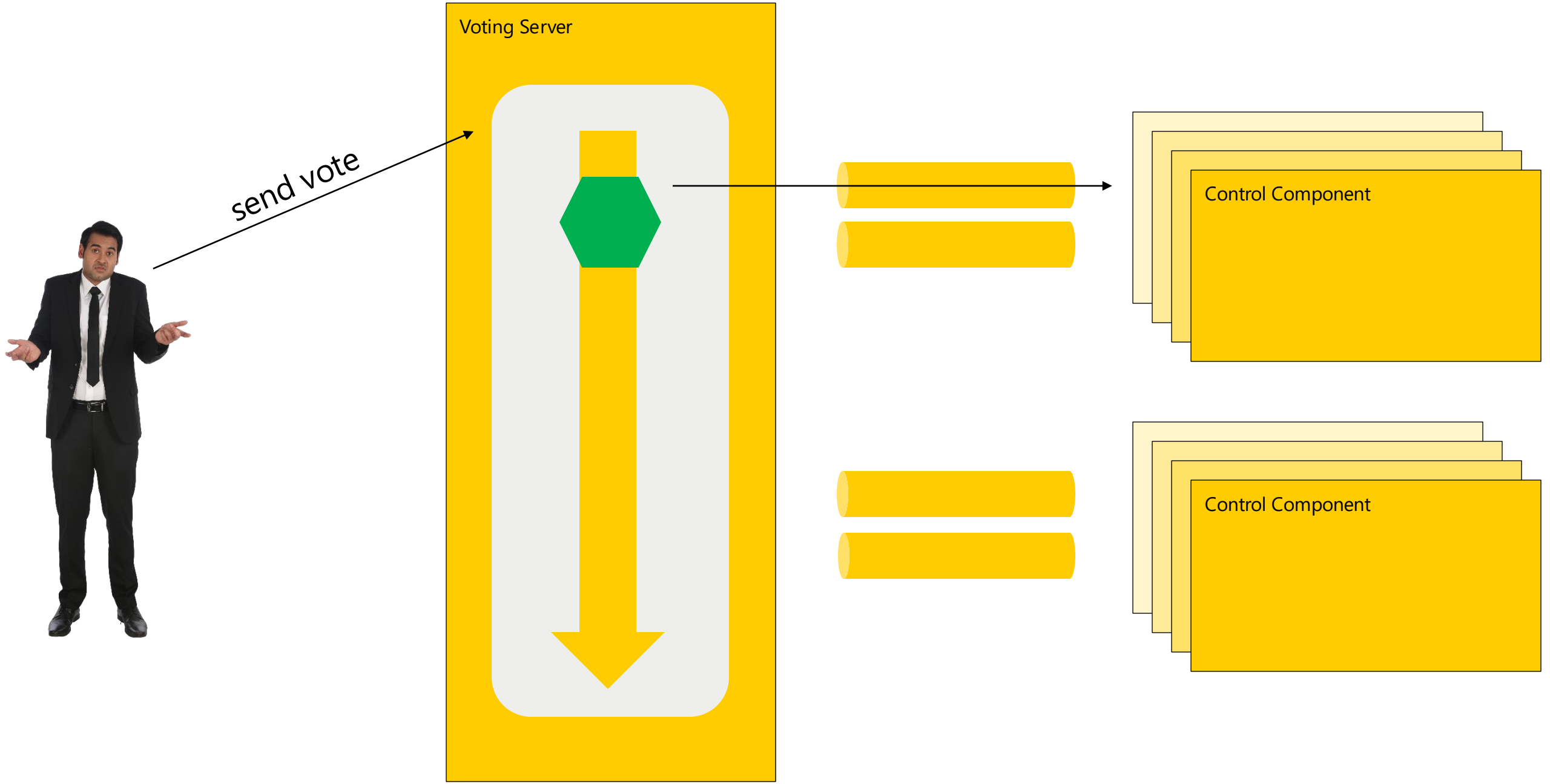
Control Component

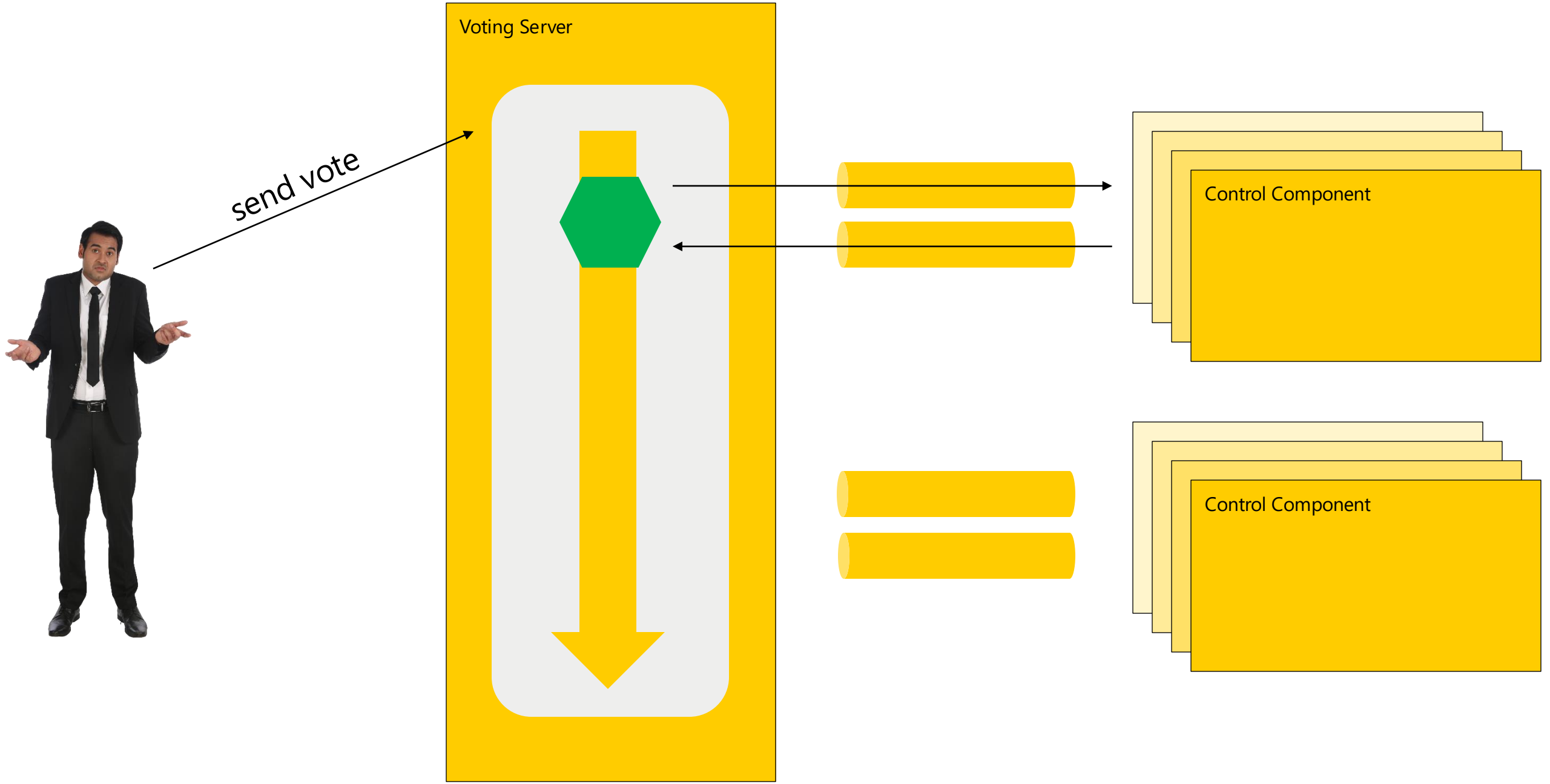
send vote

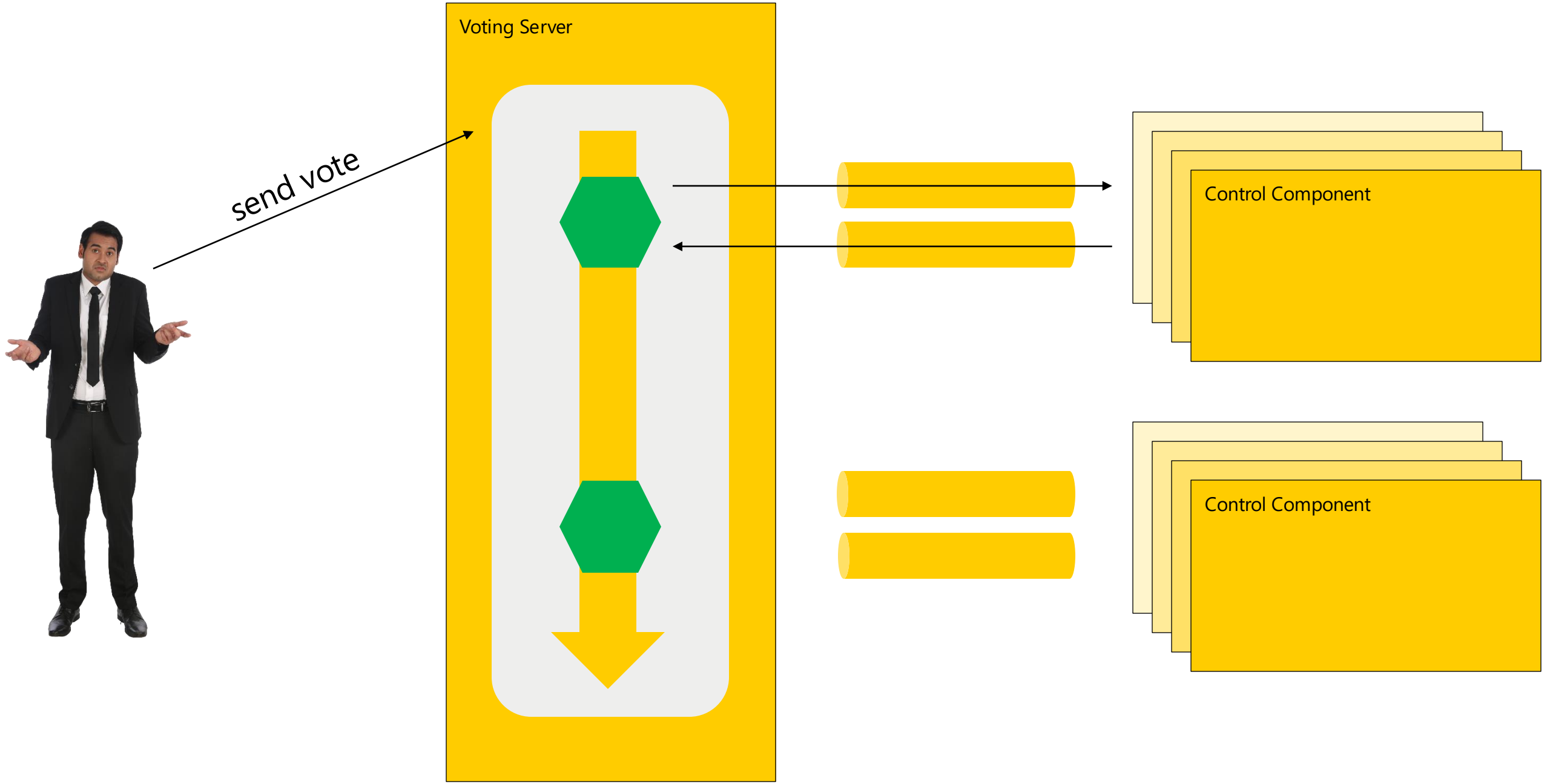
Voting Server

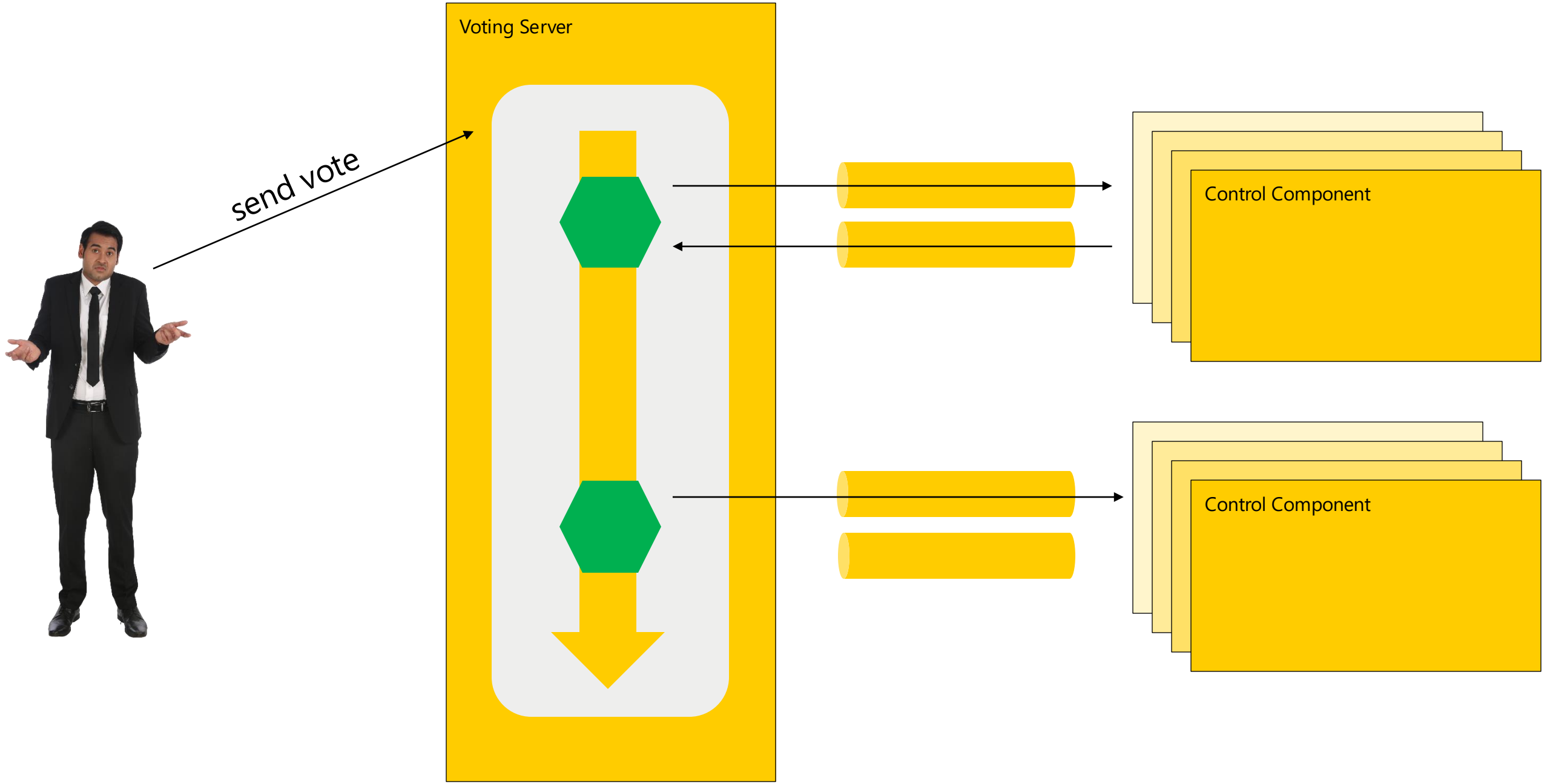
Control Component

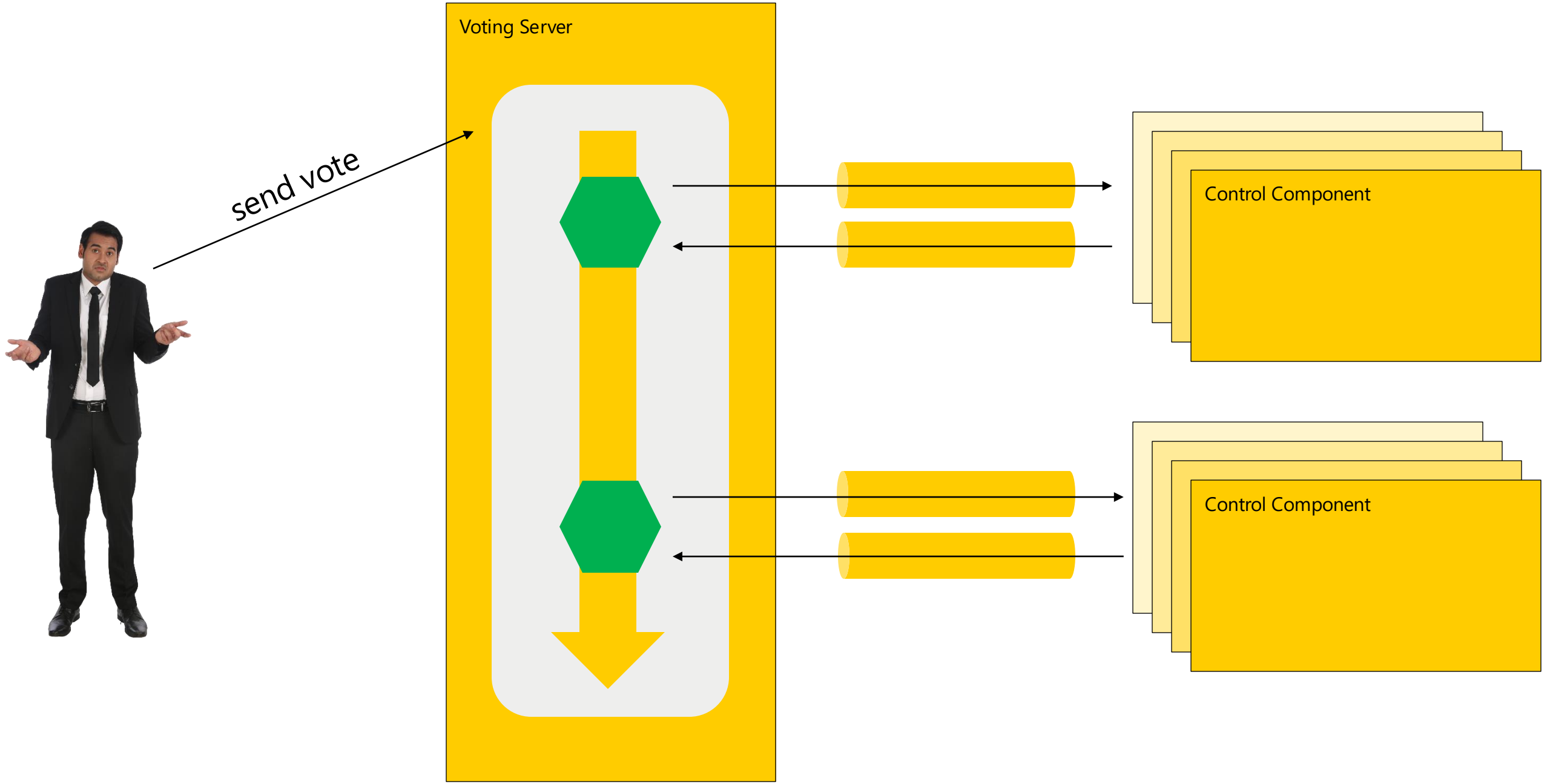
Control Component

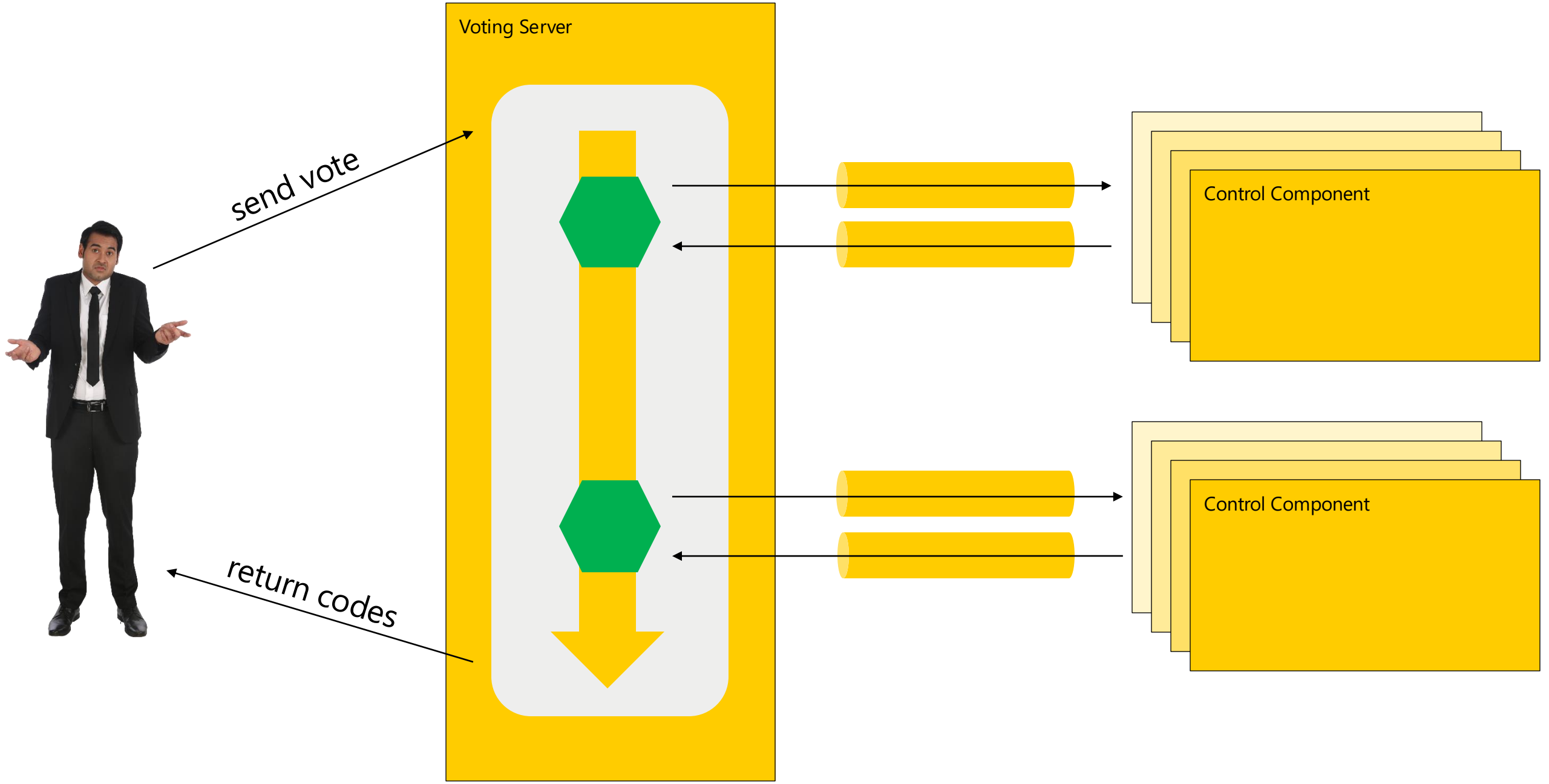

Voting Server
send vote
Control Component
Control Component


Voting Server
send vote
Control Component
Control Component


send vote
Voting Server
Control Component
Control Component


Voting Server
send vote
Control Component
Control Component


send vote
Voting Server
Control Component
Control Component


Voting Server
send vote
return codes
Control Component
Control Component

Voting Server

Control Component

Control Component

send vote

Voting Server

Control Component

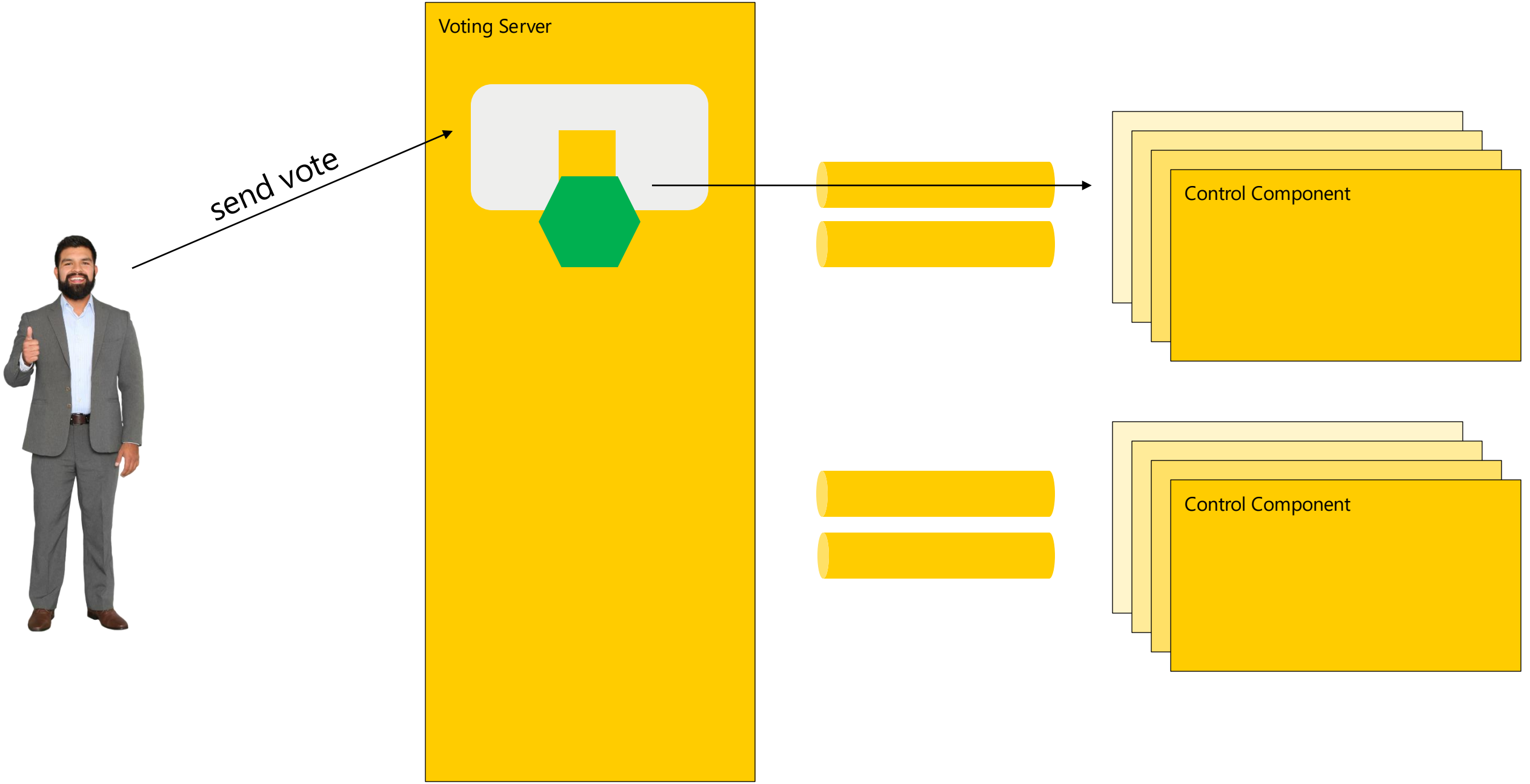
Control Component

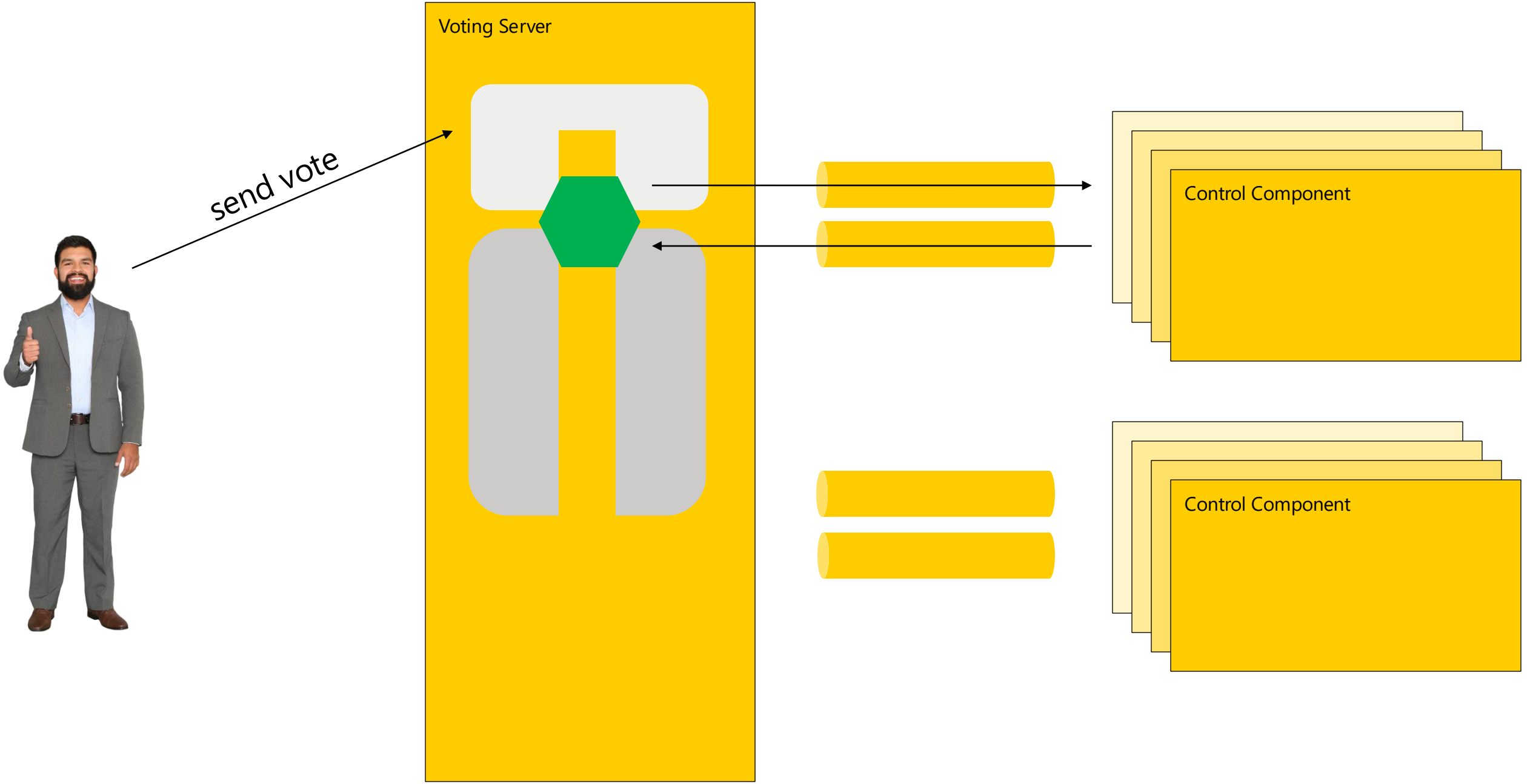
send vote

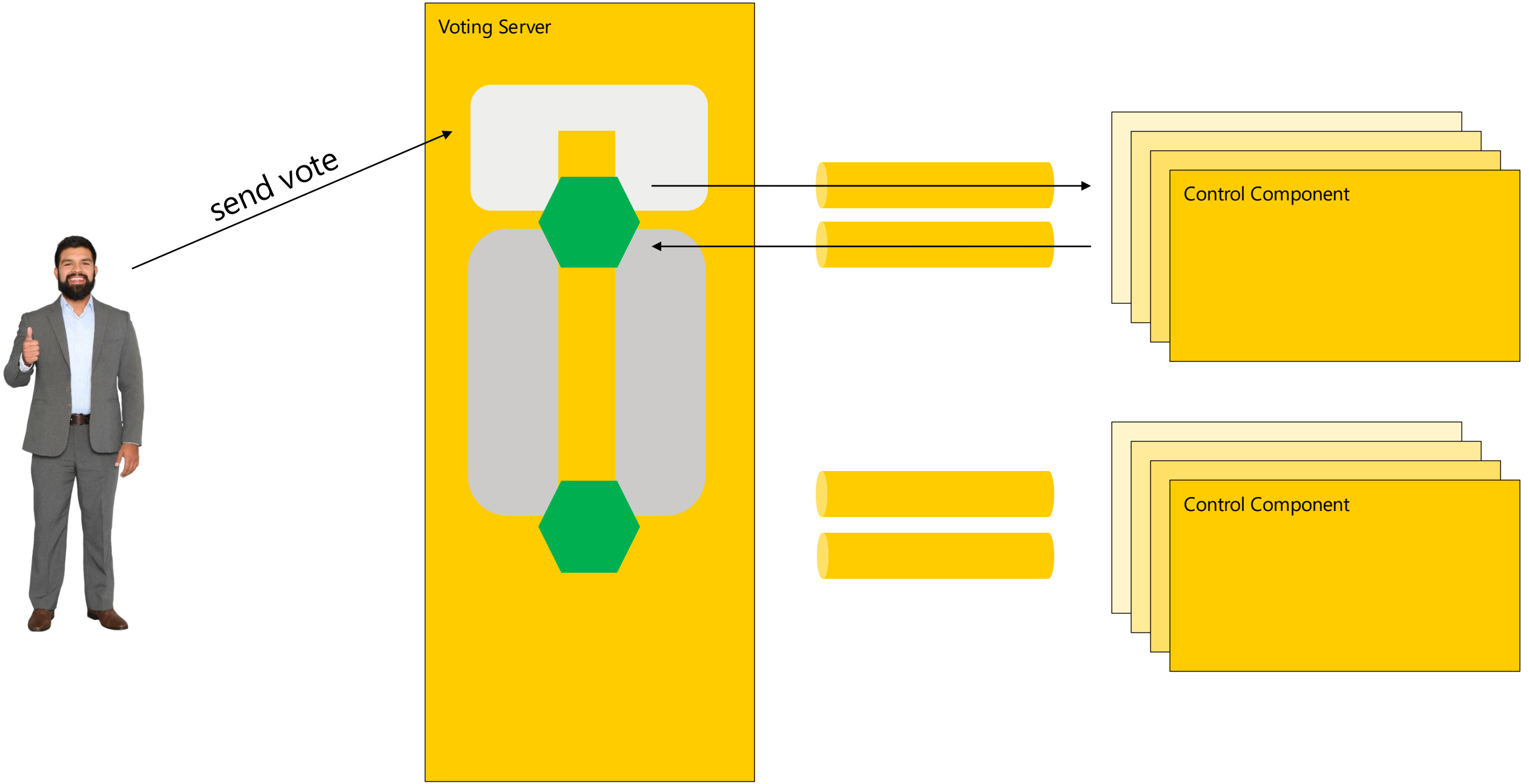
Voting Server

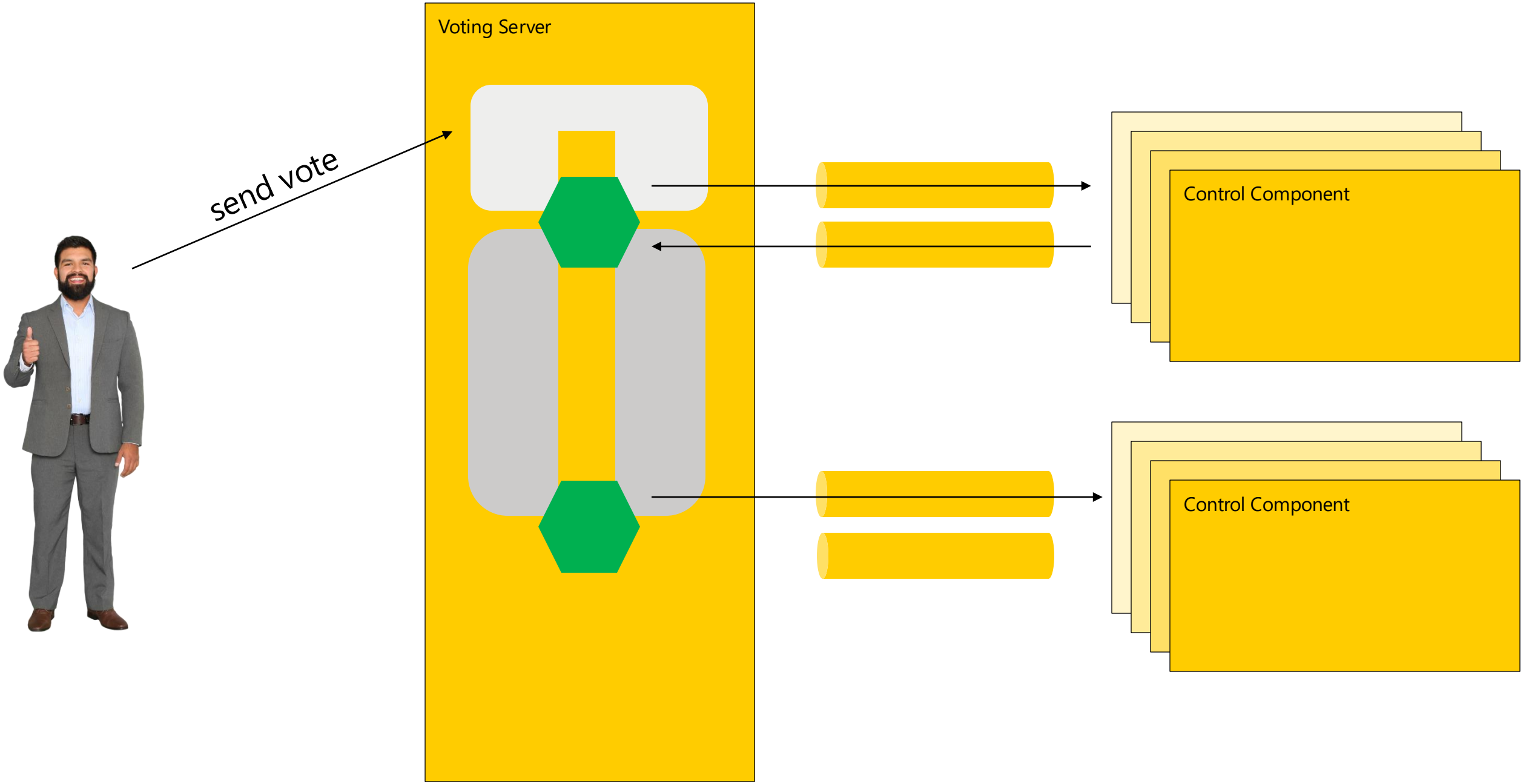
Control Component

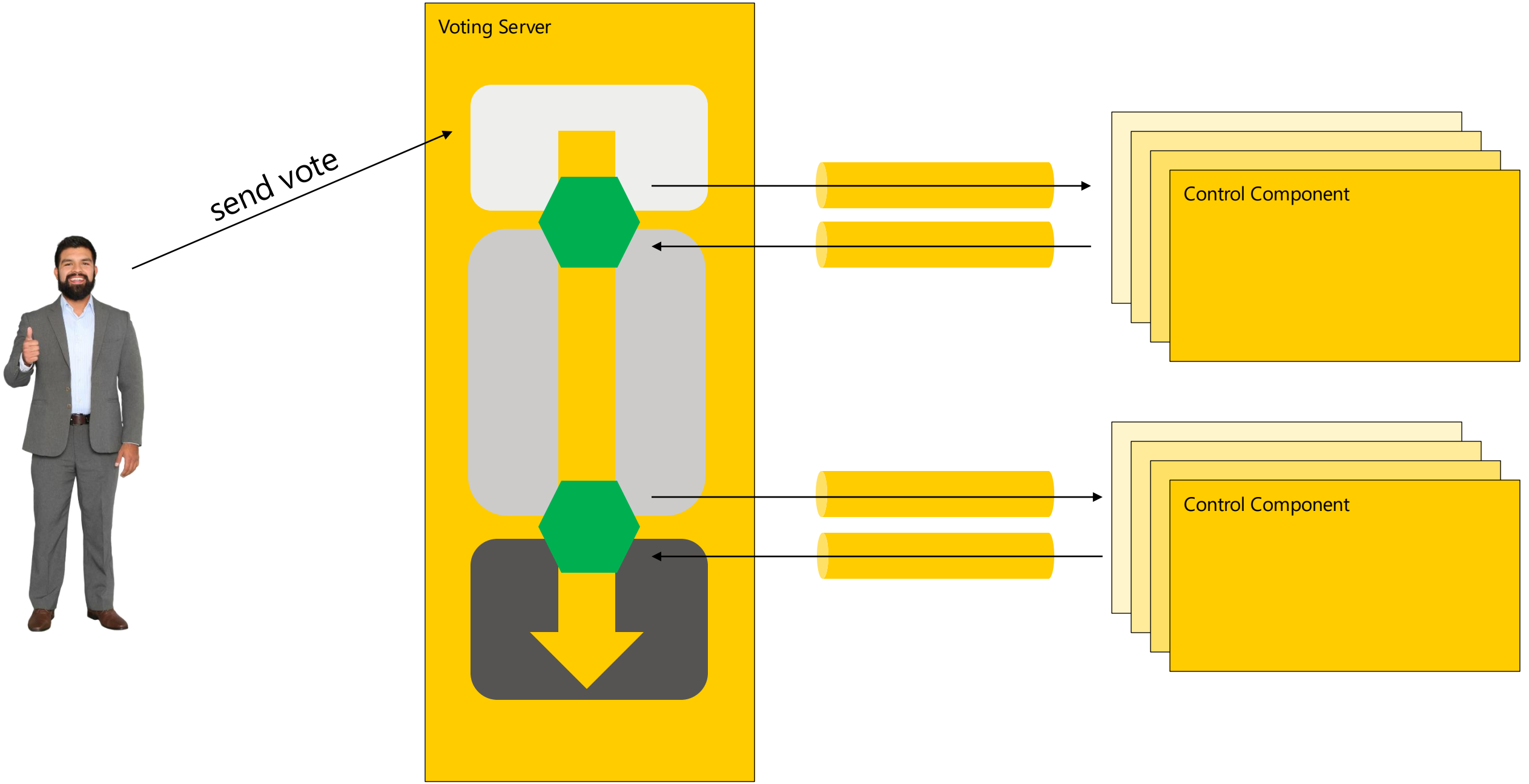
Control Component

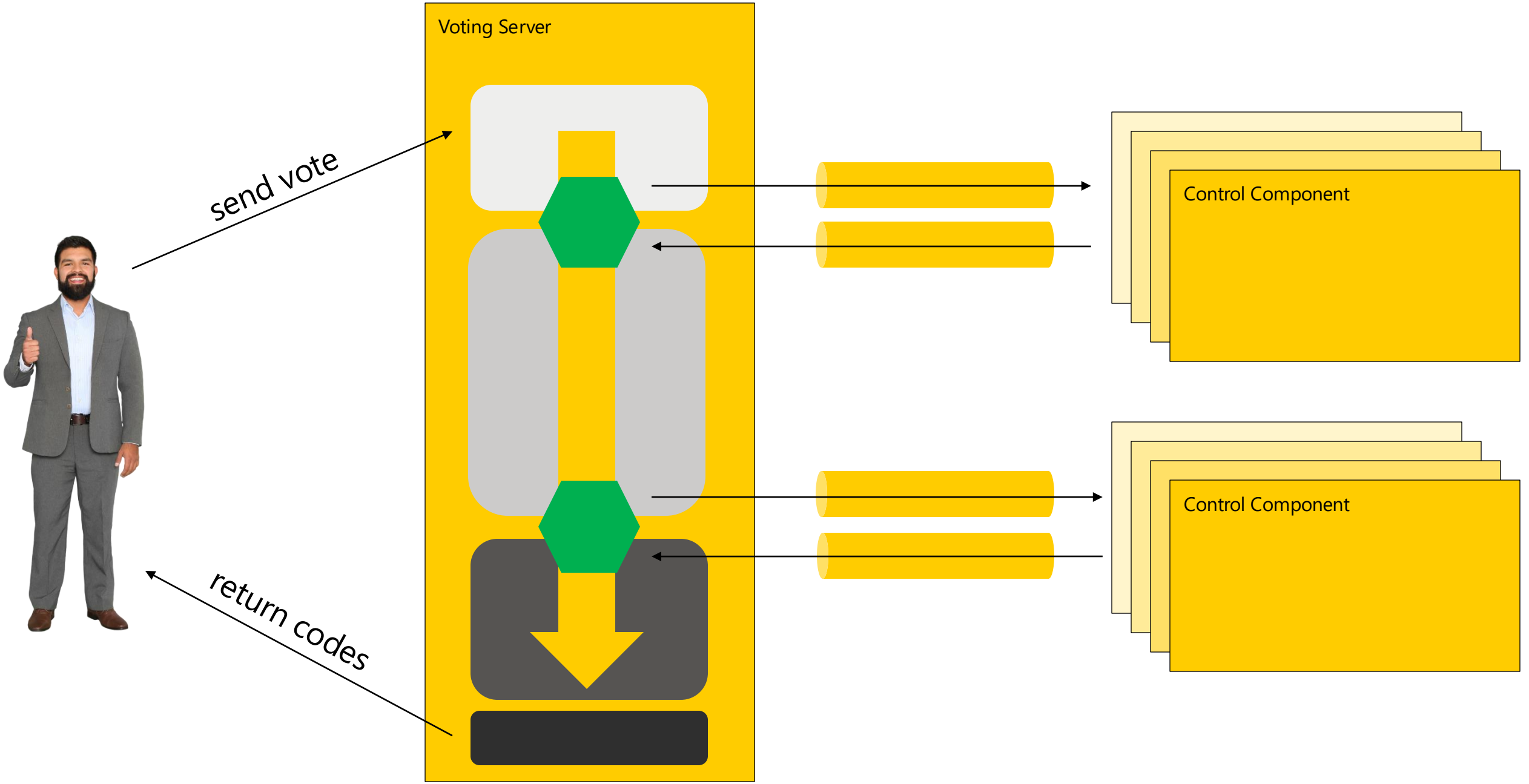

Voting Server
send vote
Control Component
Control Component


Voting Server
send vote
Control Component
Control Component


send vote
Voting Server
Control Component
Control Component


send vote
Voting Server
Control Component
Control Component


send vote
Voting Server
Control Component
Control Component


Voting Server
send vote
return codes
Control Component
Control Component

Key Technical Aspects

How to?

Libraries

RabbitMQ, ActiveMQ, etc.

Threads

Async Publish/Subscribe Pattern

Queuing

Pros

Broadcast, Listener, Load Balancing

Cons

Possible duplicate messages

3 Key Takeaways

🎯 Objectives

- Scalability
- Availability
- Prevent timeouts
- Transactional (stable states)

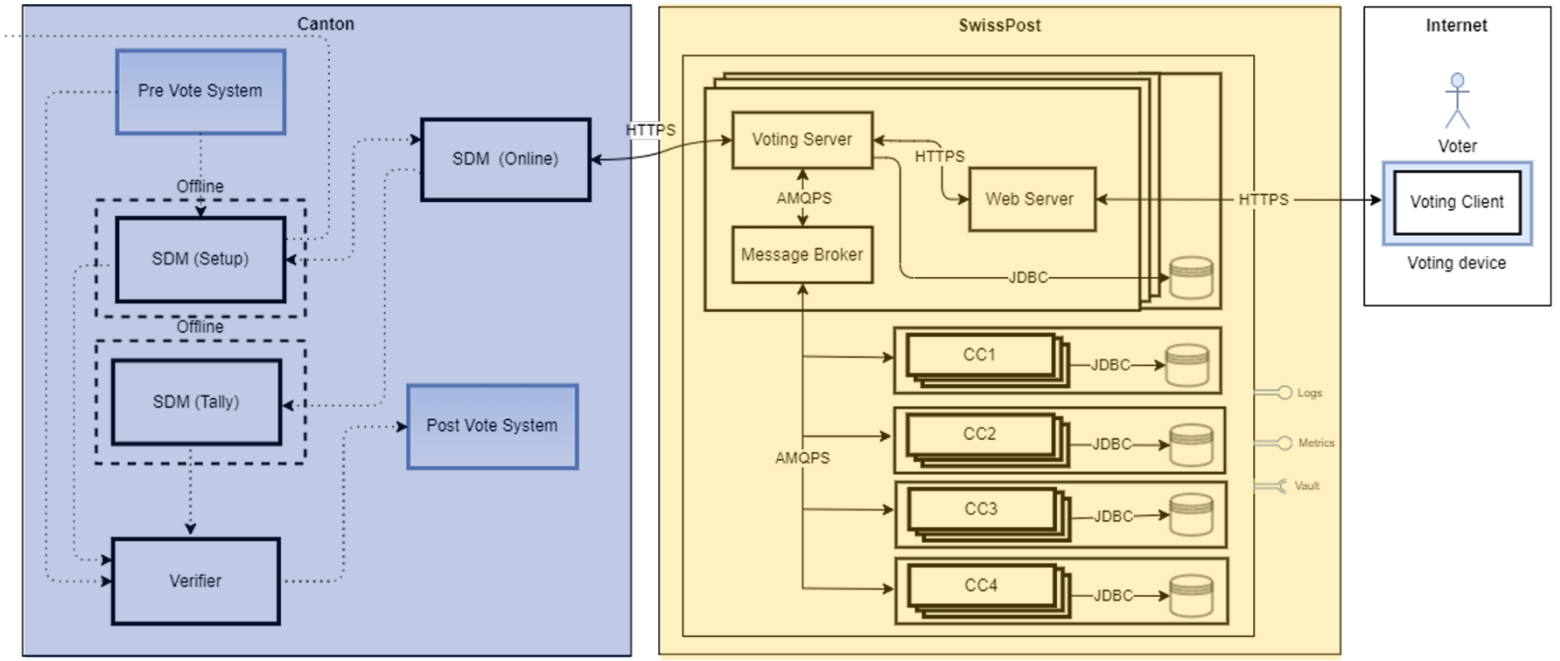
⚖️ Trade-offs

- Increased complexity
- Throughput vs. delay
- Consistency vs. availability
- Operational overhead

Interactions between systems must be considered as soon as possible

Challenges

1. **Scalability**
2. **Data transfer**
3. **Resiliency**



Want to know more?

Check out our **published code** on GitLab and test the system

Submit a report to our **bug bounty programme** in YesWeHack
💰 up to € 230,000

Experience E-Voting with the **demo**

Check the Swiss Post **opportunities** during and after studies

Thanks!

Céline Camacho
E-Voting Software Engineer

celine.camacho@post.ch

IN’21

Nils Aellen
E-Voting Solution Architect

nils.aellen@post.ch

IN’02

Patrick Andrade
E-Voting People DevOps Lead

patrick.oliveiraandrade@post.ch

IN’17