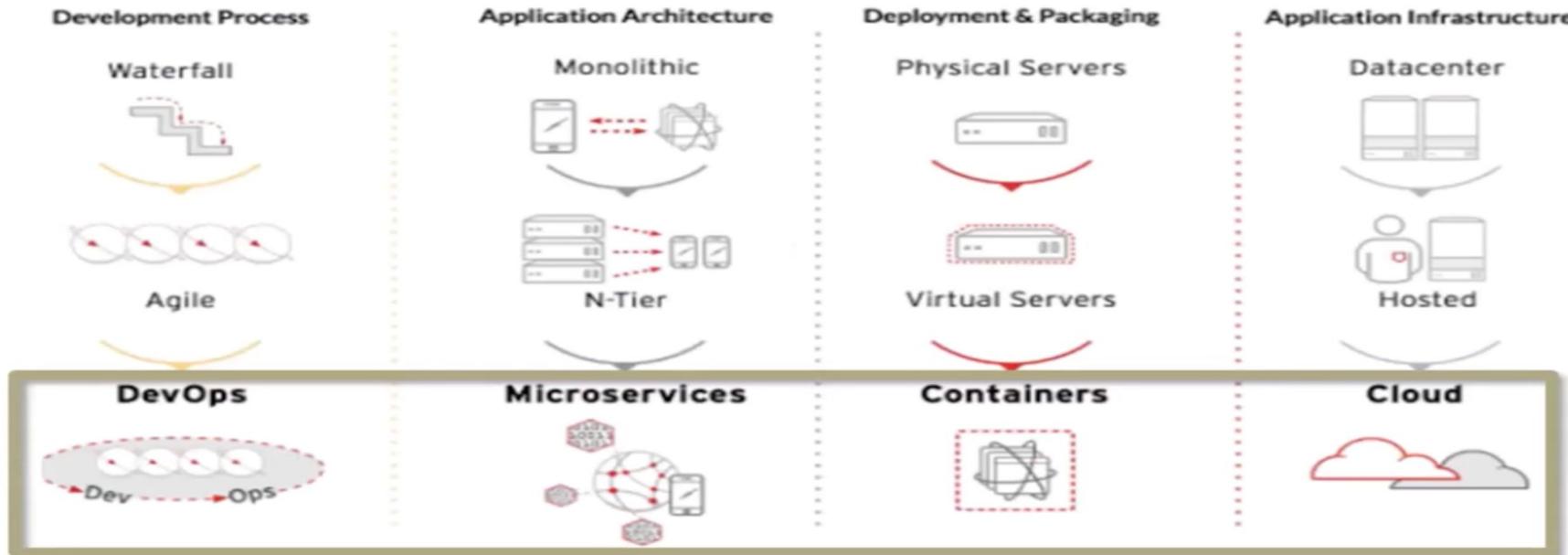
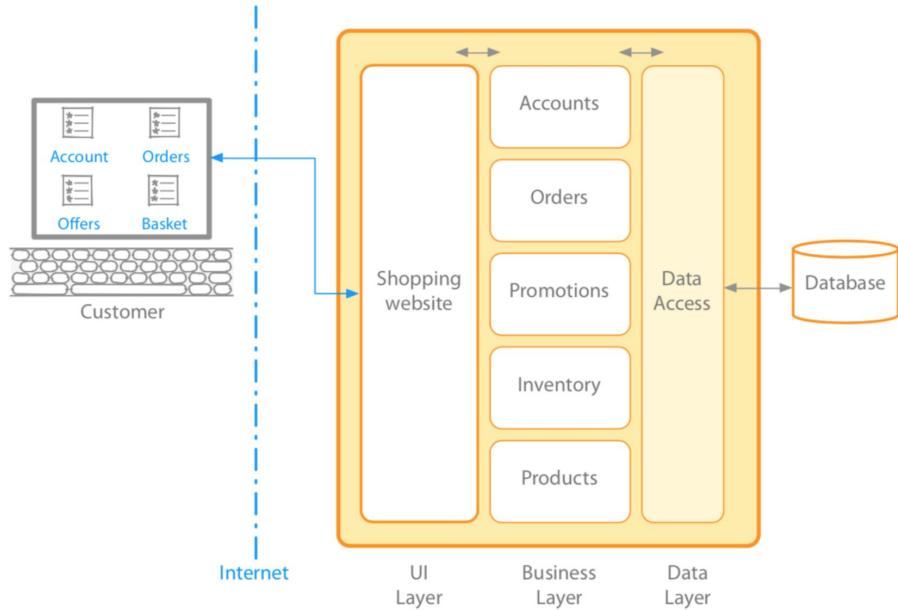


Software Engineering evolution



From Monolithic To Microservices

Monolithic architecture - Ex



Monolithic architecture - pros

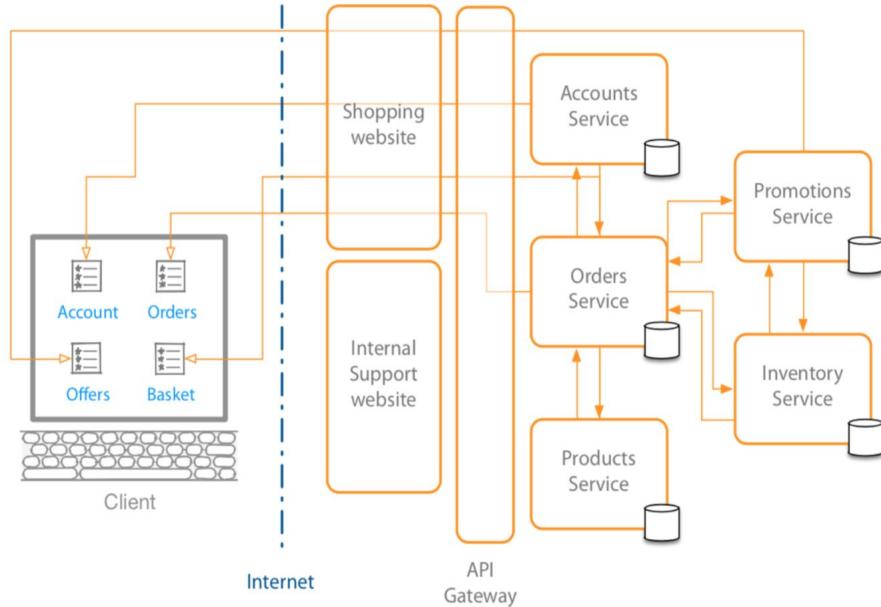
- Simpler development & deployment
- Fewer cross-cutting concerns
- Better performance

Monolithic architecture - cons

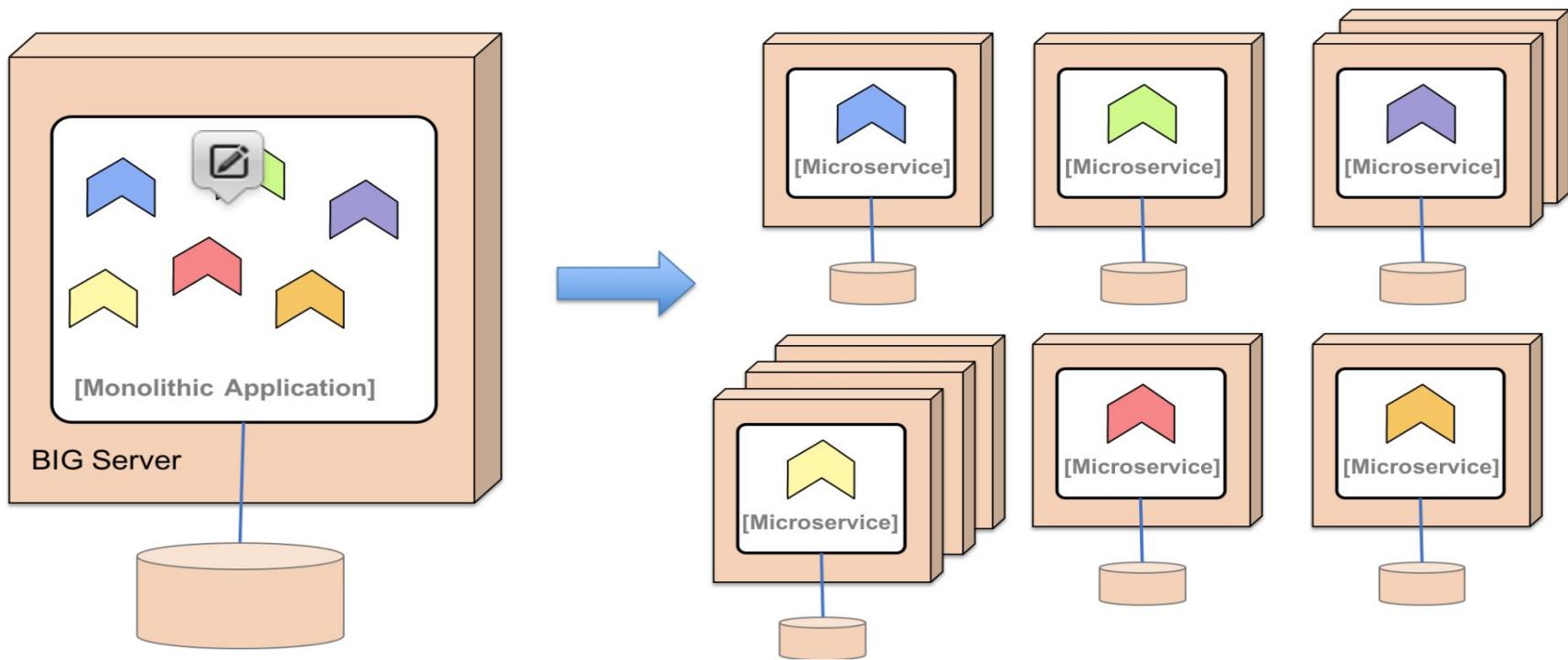
- Larger codebase
- Difficult to adopt new technologies
- Failure could affect whole system
- Scaling requires duplication of the whole
- Limited agility
 - every small update requires a full redeployment
 - all developers have to wait until it's done
 - When several teams are working on the same project, agility can be reduced greatly.

Enter Microservices

Microservice - Ex



Defining a microservice

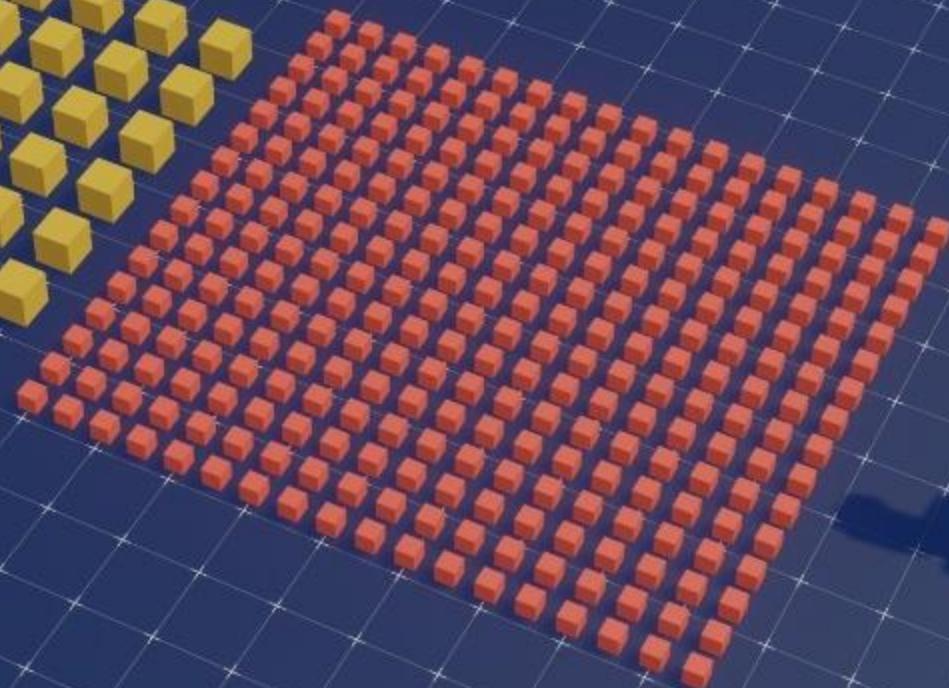
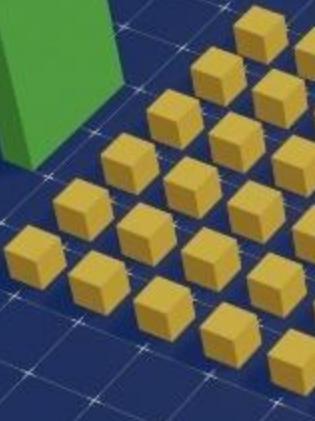


Microservice - characteristics

- Must conforms to a shared-nothing architecture
- Must only communicate with well defined interfaces (Api calls | Messages)
- Must be deployed as separate runtime process e.g Docker
- Microservice instances are stateless

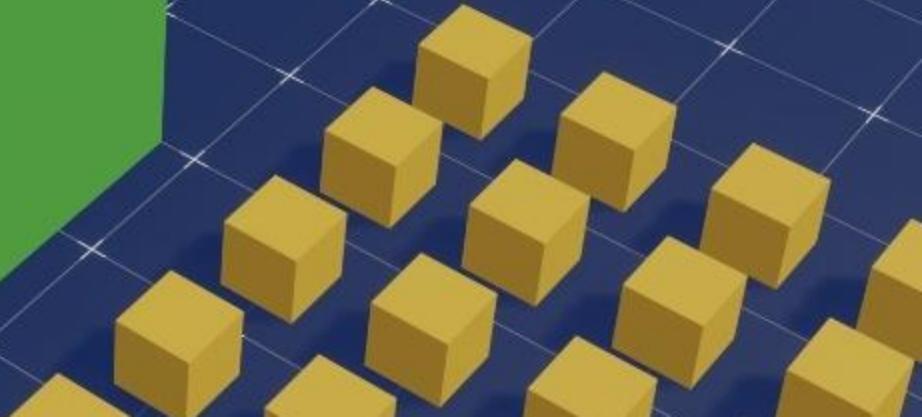
Microservice - benefits

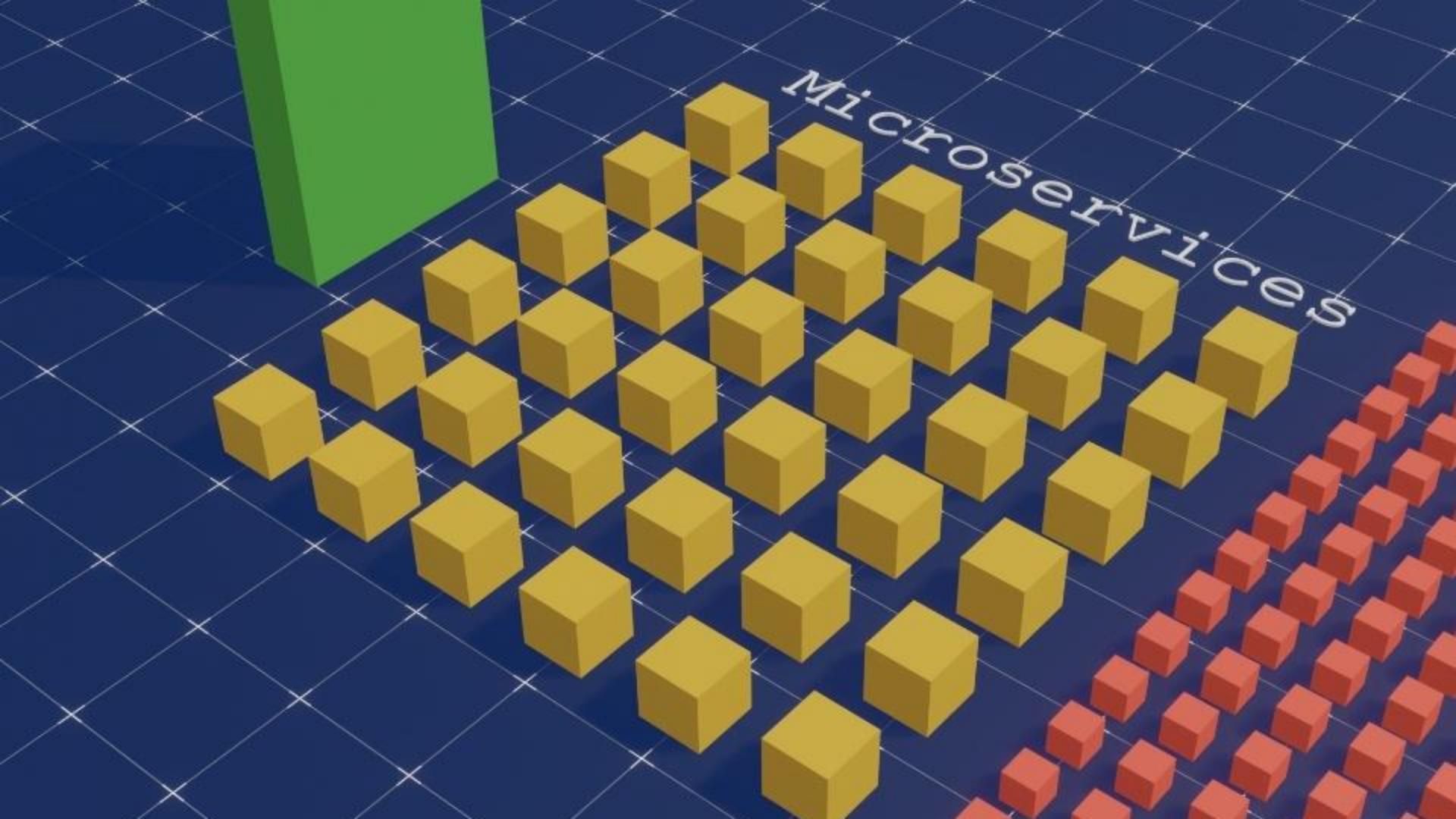
- Easy to develop,test and deploy
- Increased agility
- Ability to **scale** horizontally



???

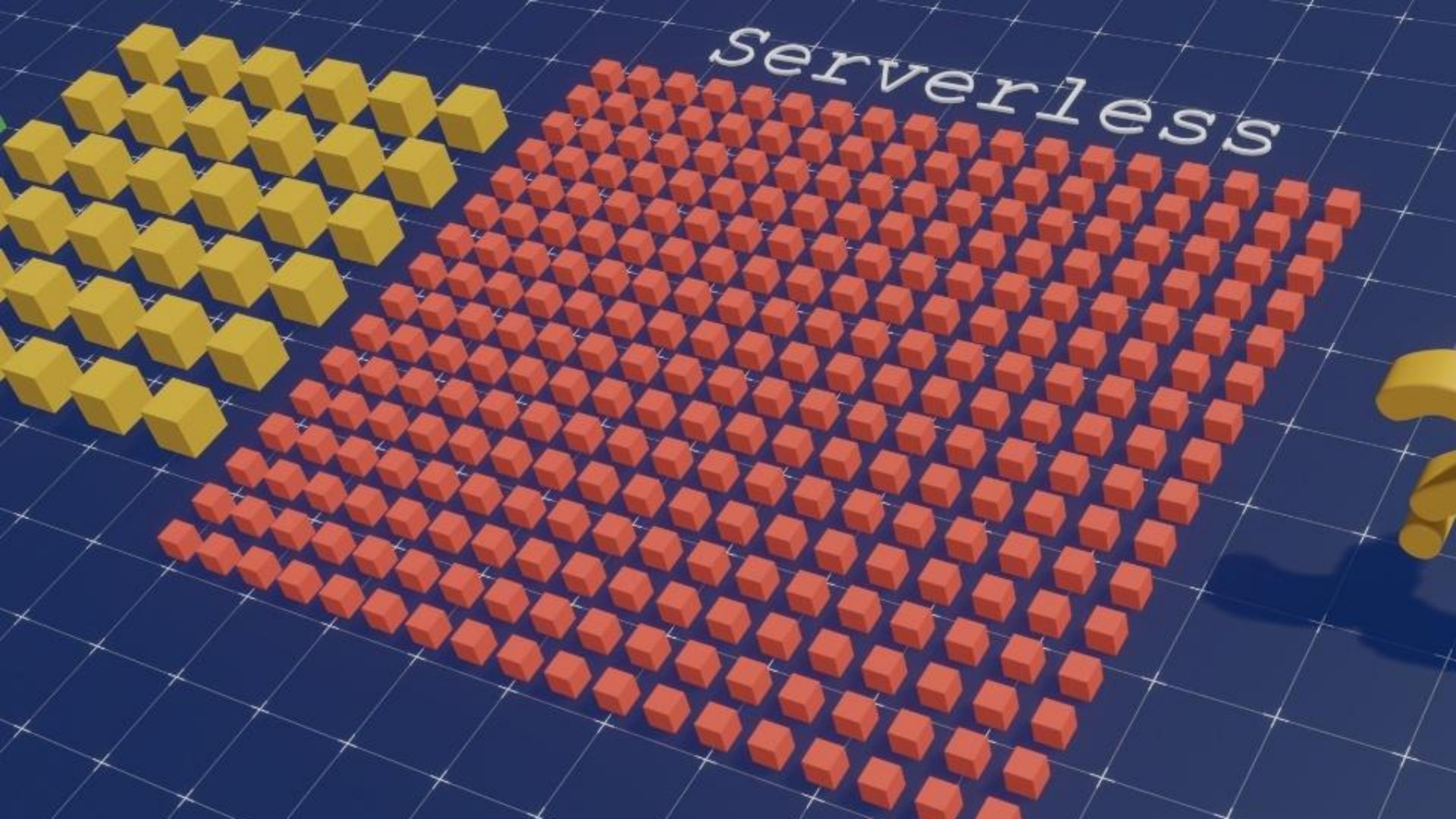
Monolith

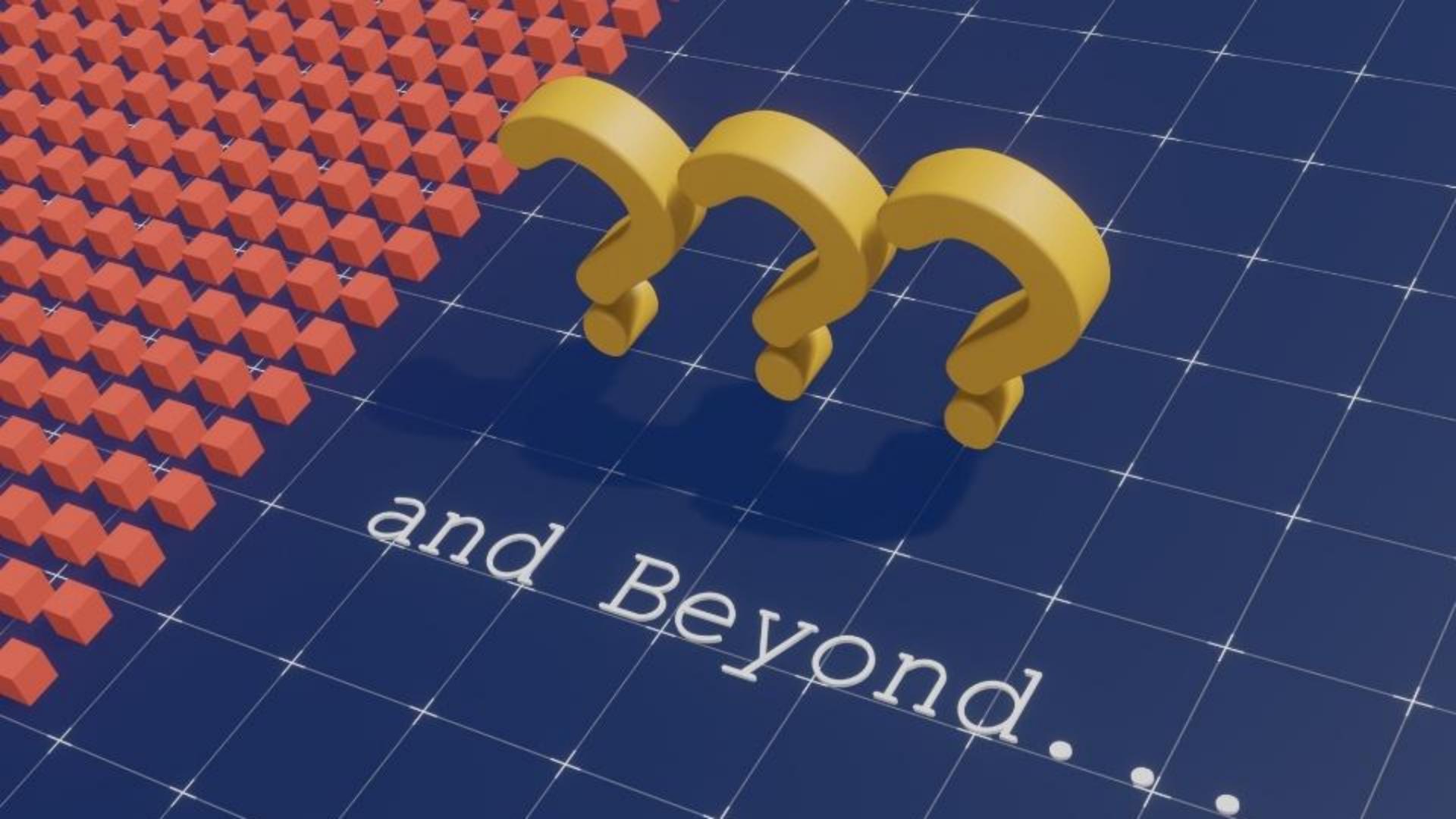




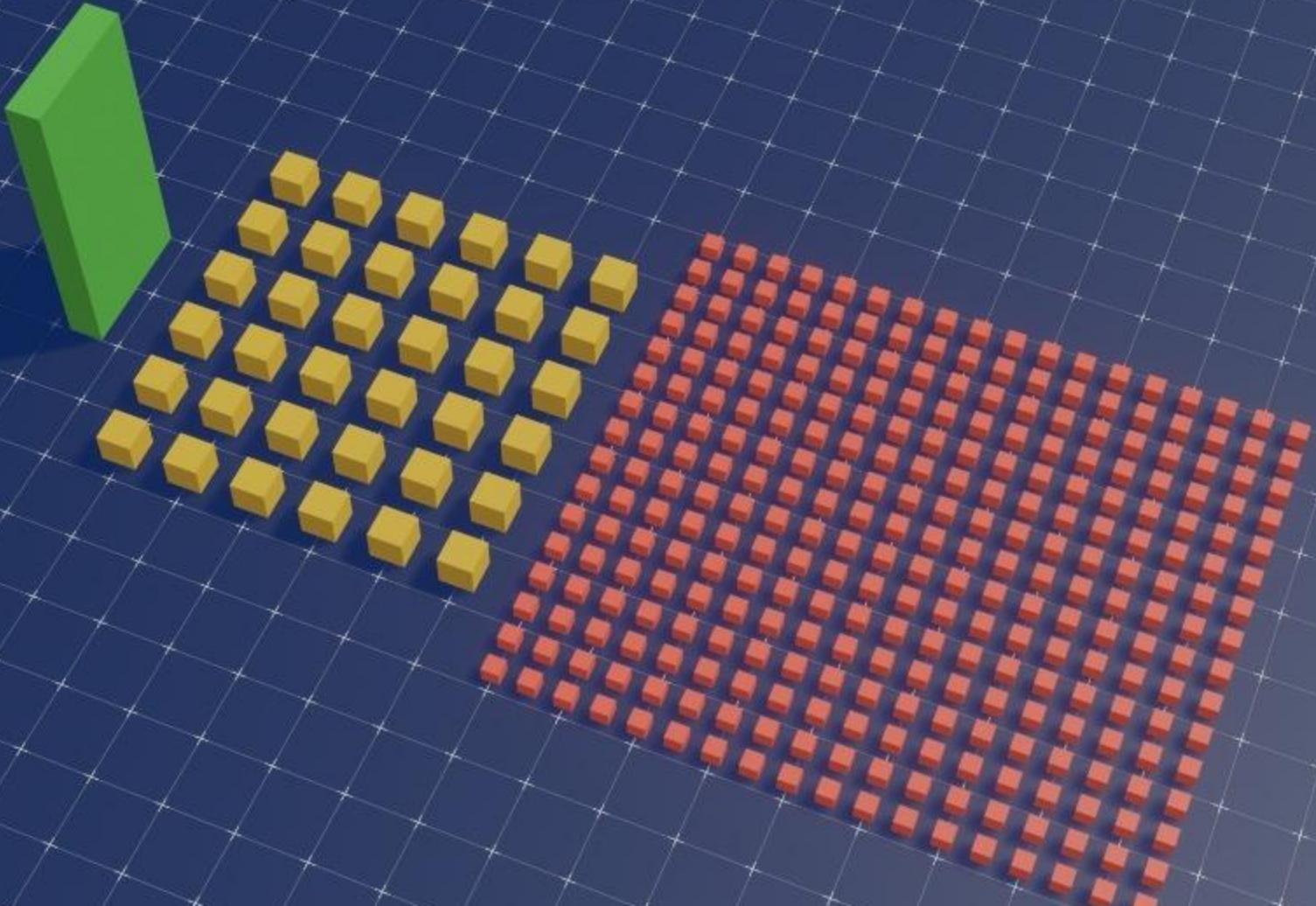
Microservices

Serverless

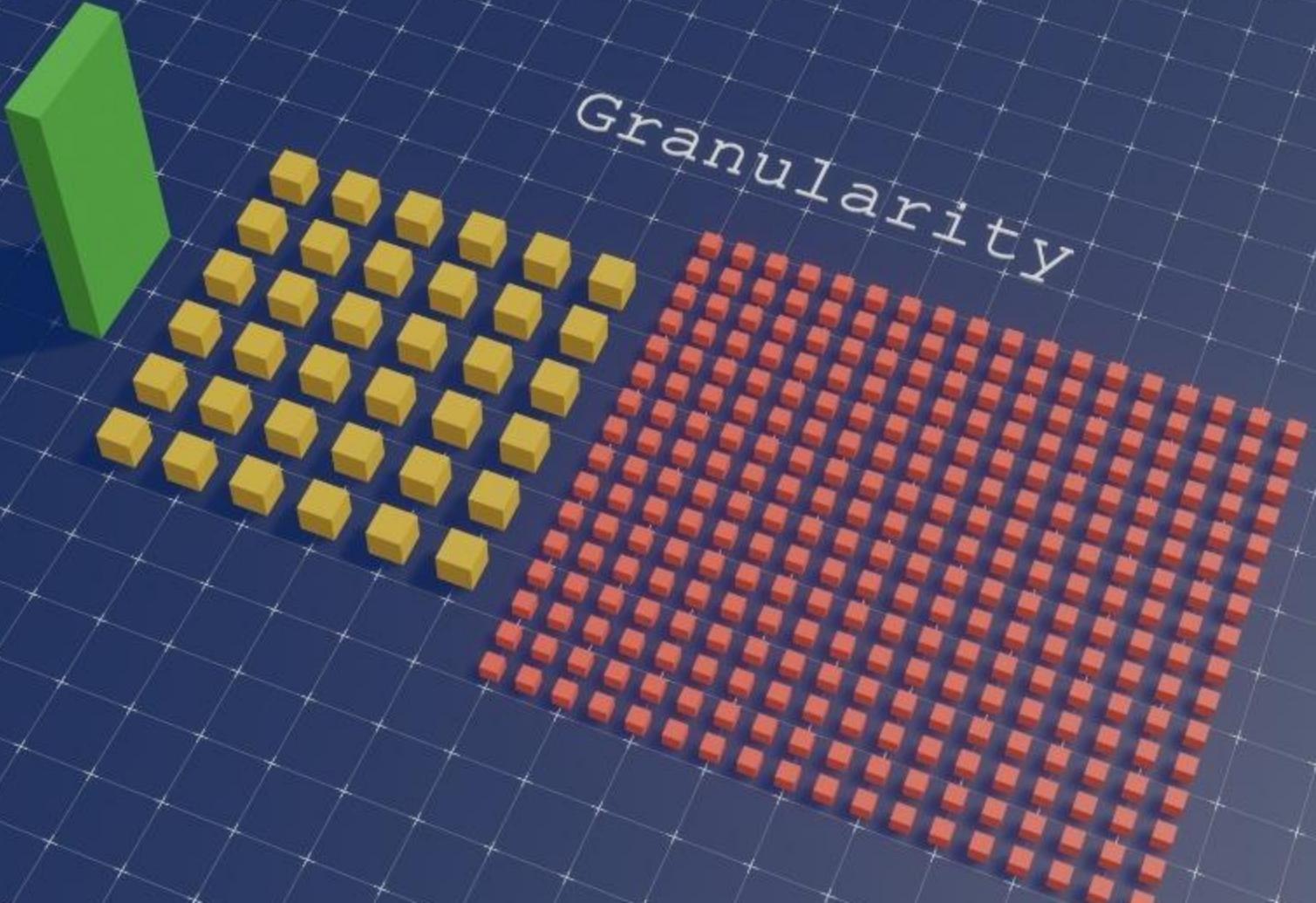




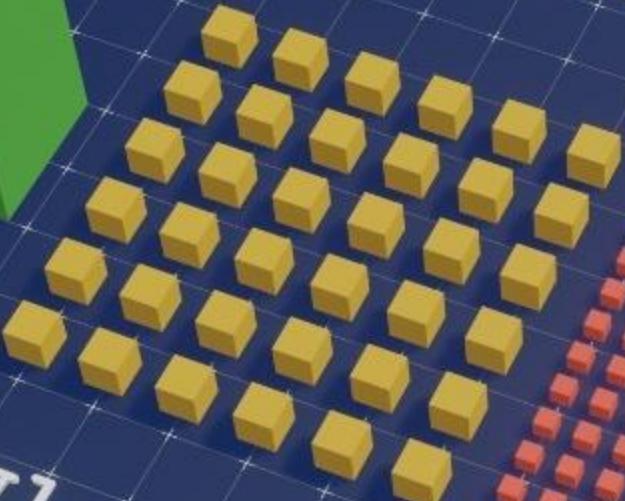
and Beyond.



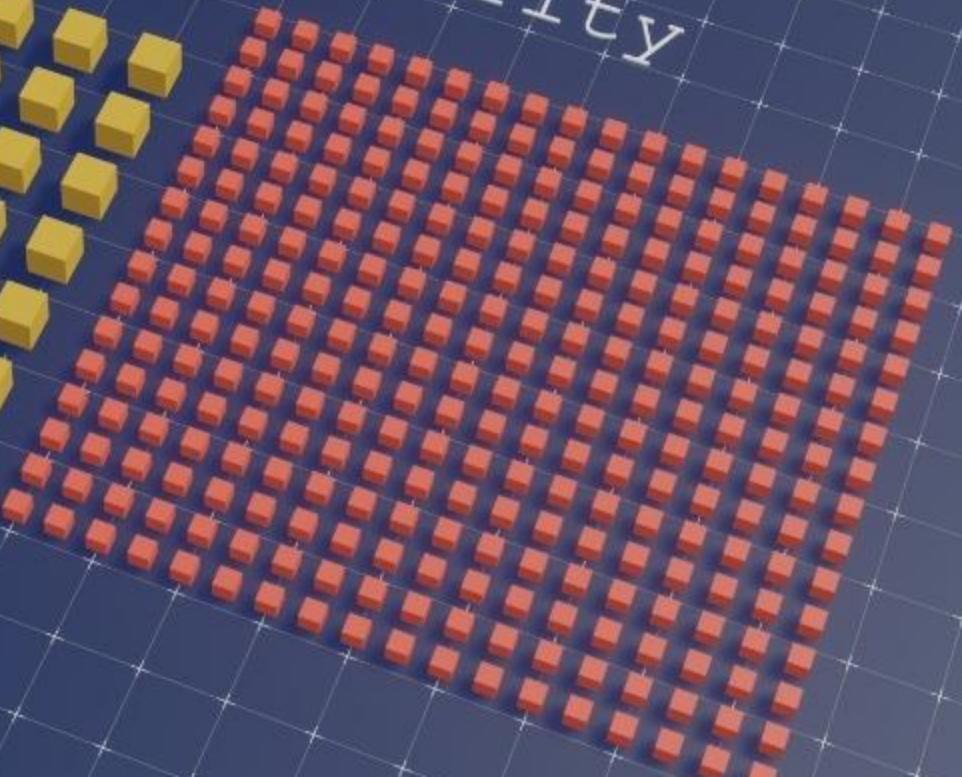
Granularity



Why?

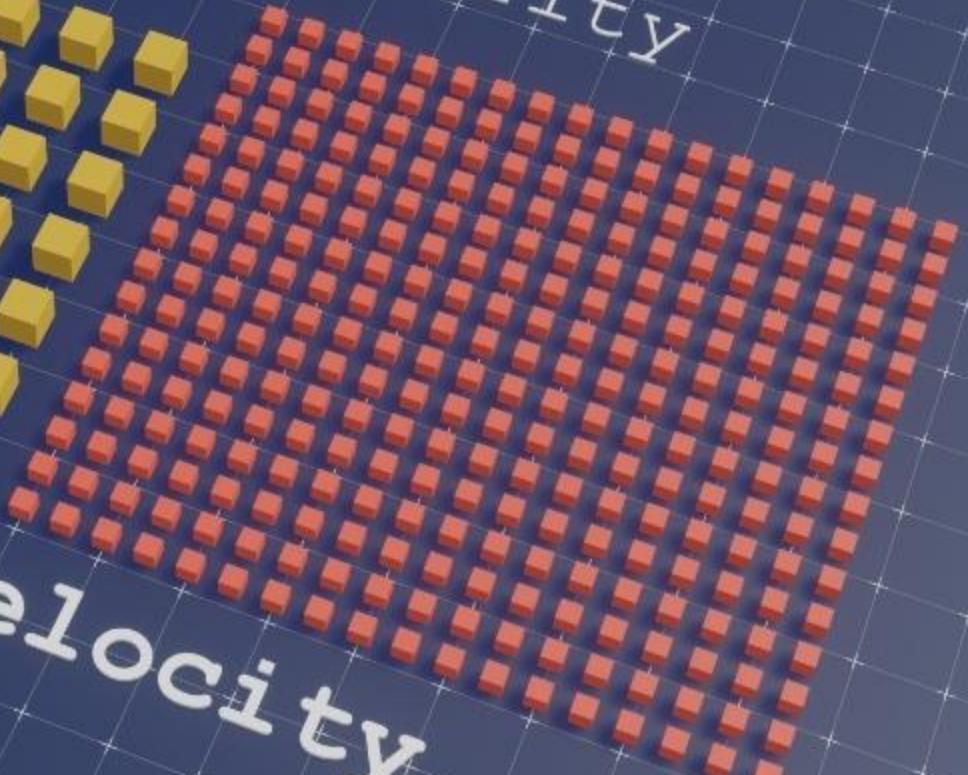
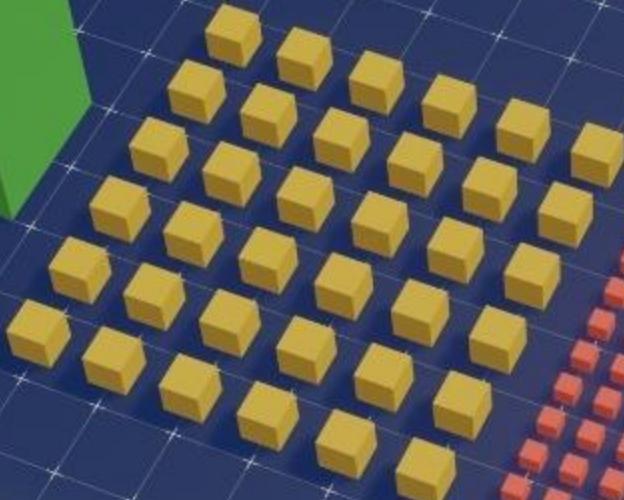


Granularity

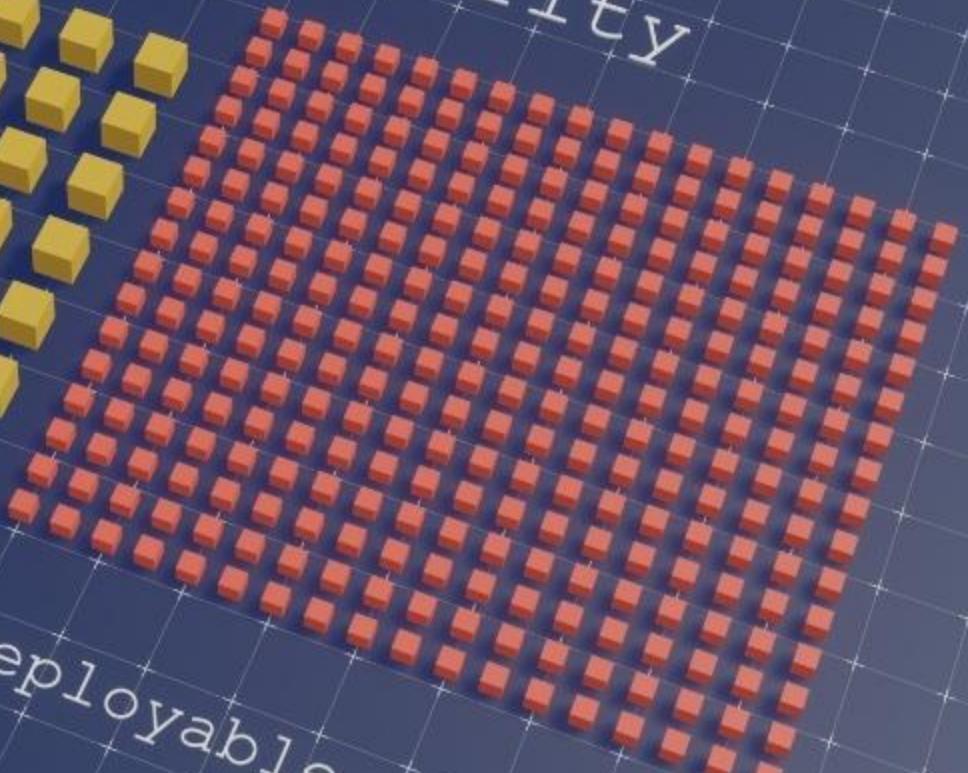
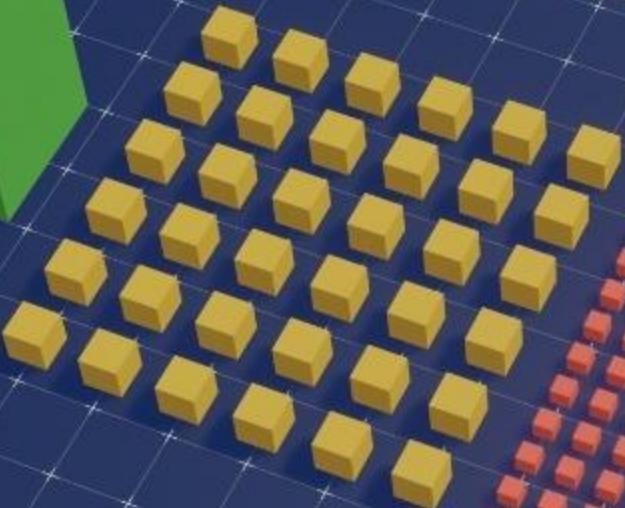


Innovation velocity

Granularity



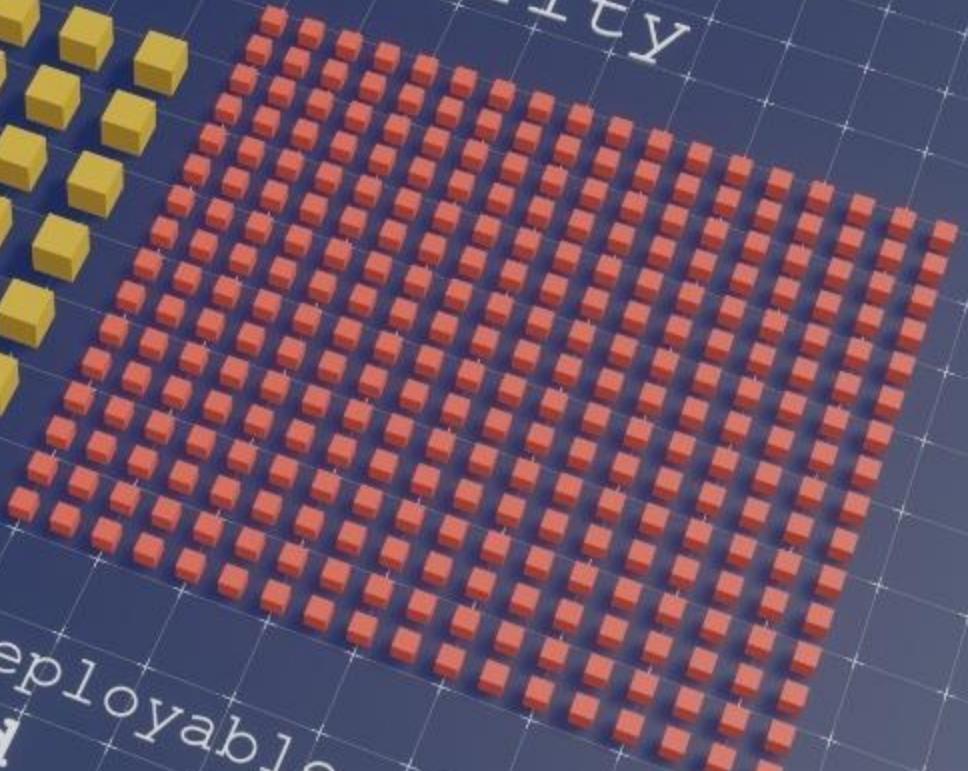
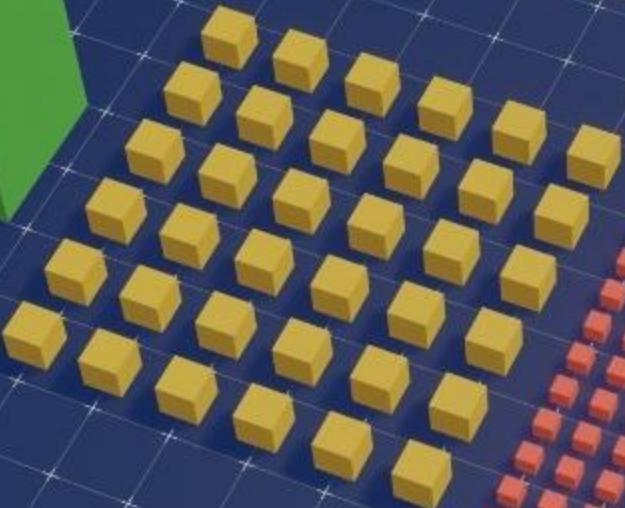
Granularity



independently deployable

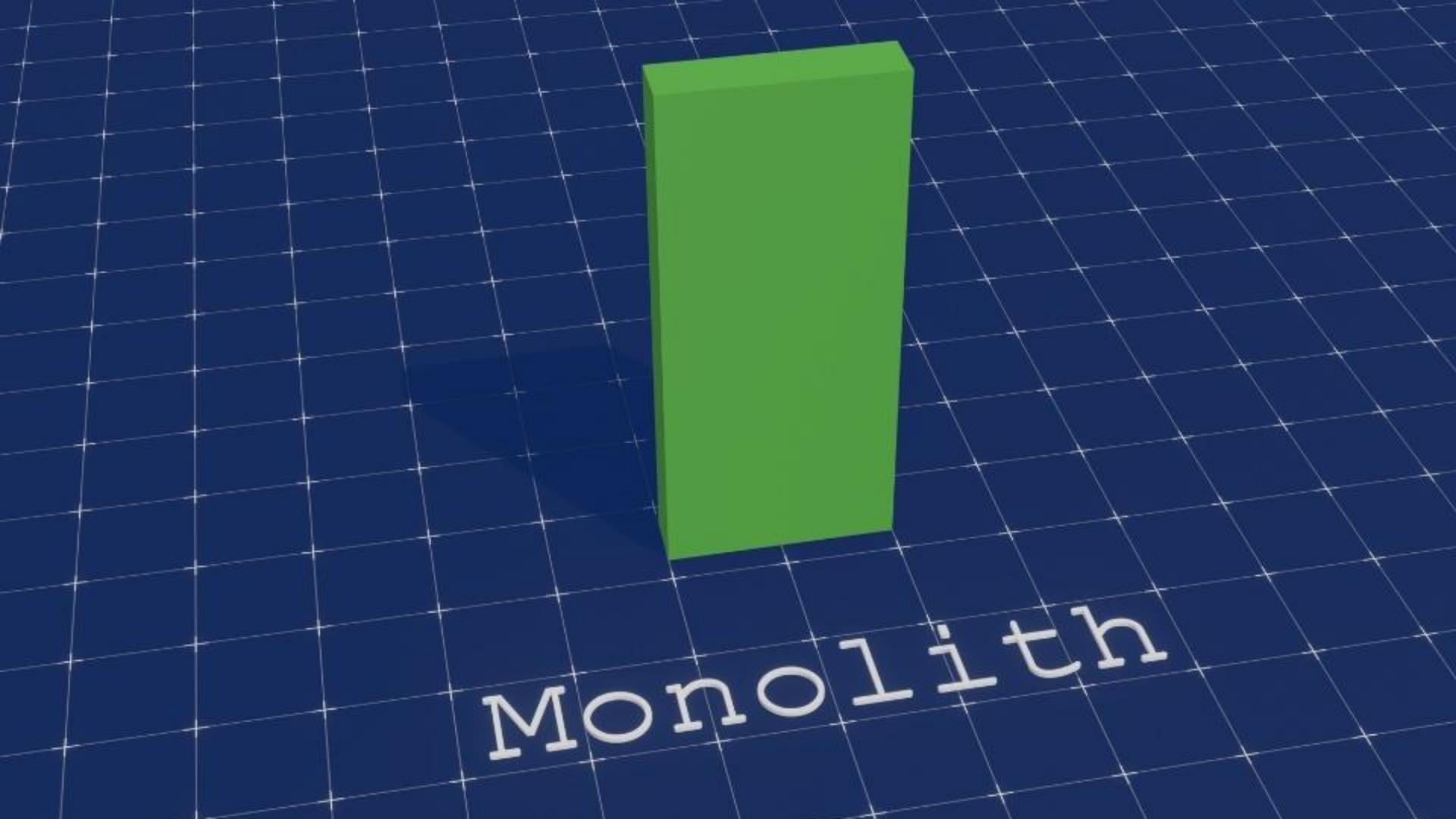


Granularity

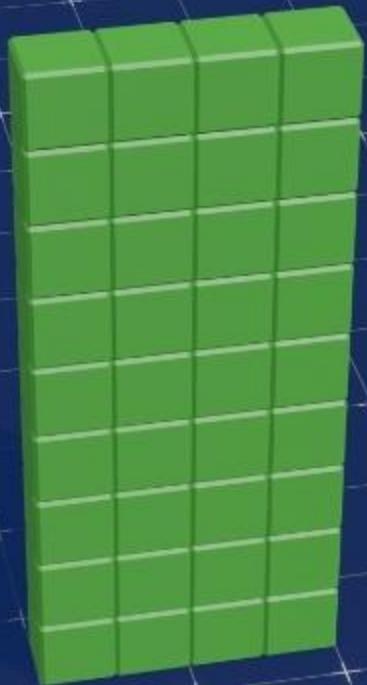


*independently deployable
loosely coupled*

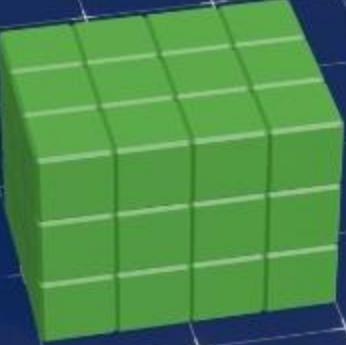


A large, solid green rectangular prism stands vertically on a blue background that features a white square grid. The word "Monolith" is written in a white, sans-serif font, positioned at the base of the green block and angled upwards towards the right.

Monolith

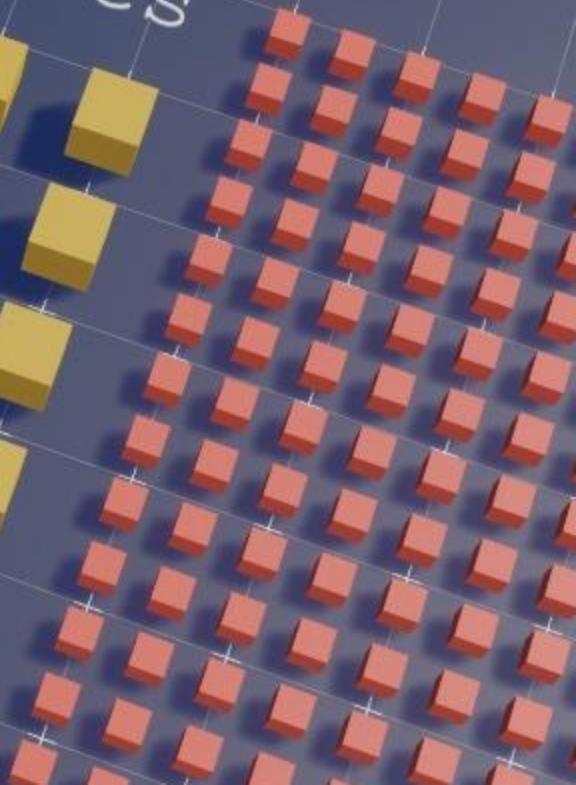
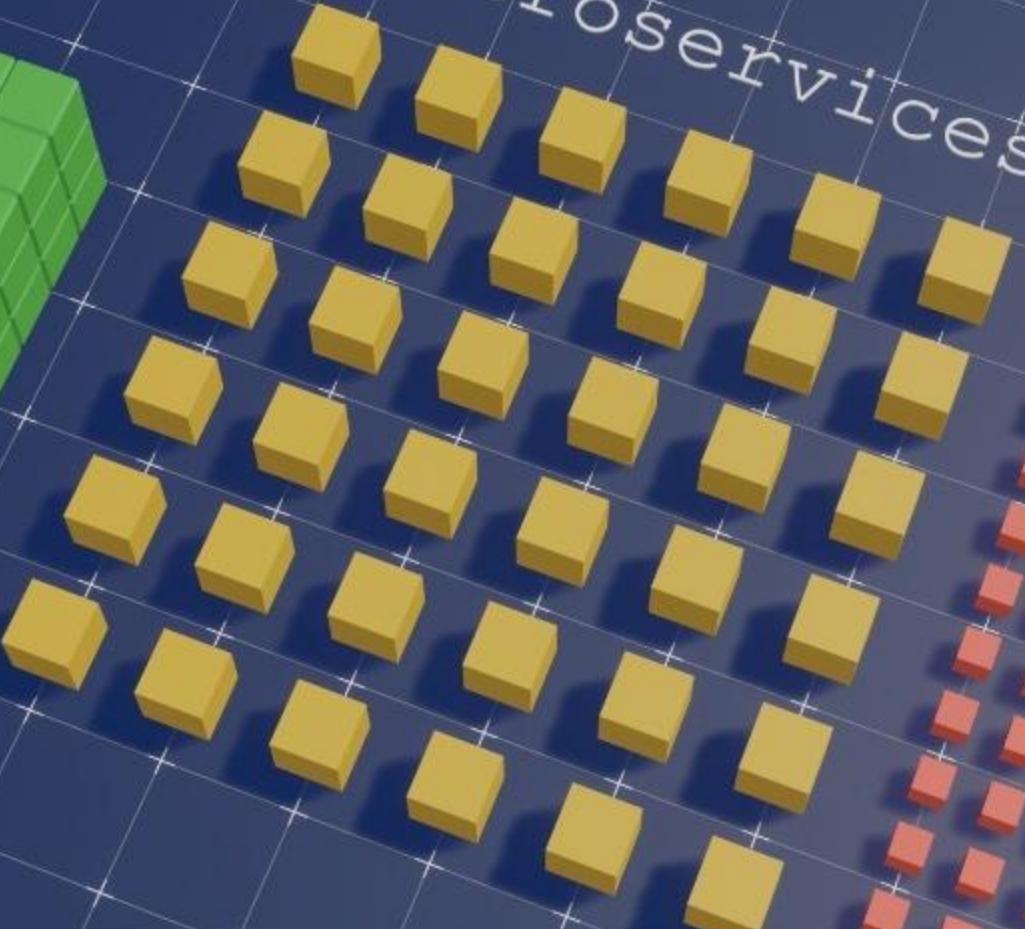


Monolith
Modules

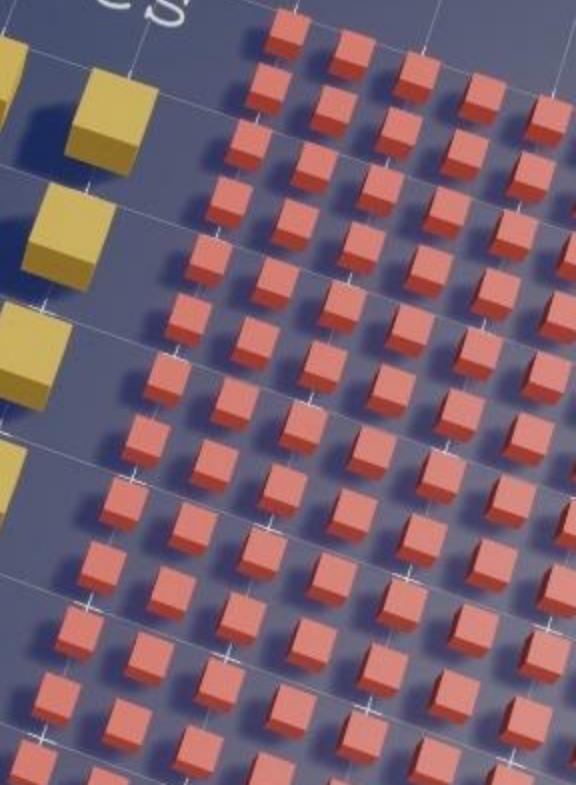
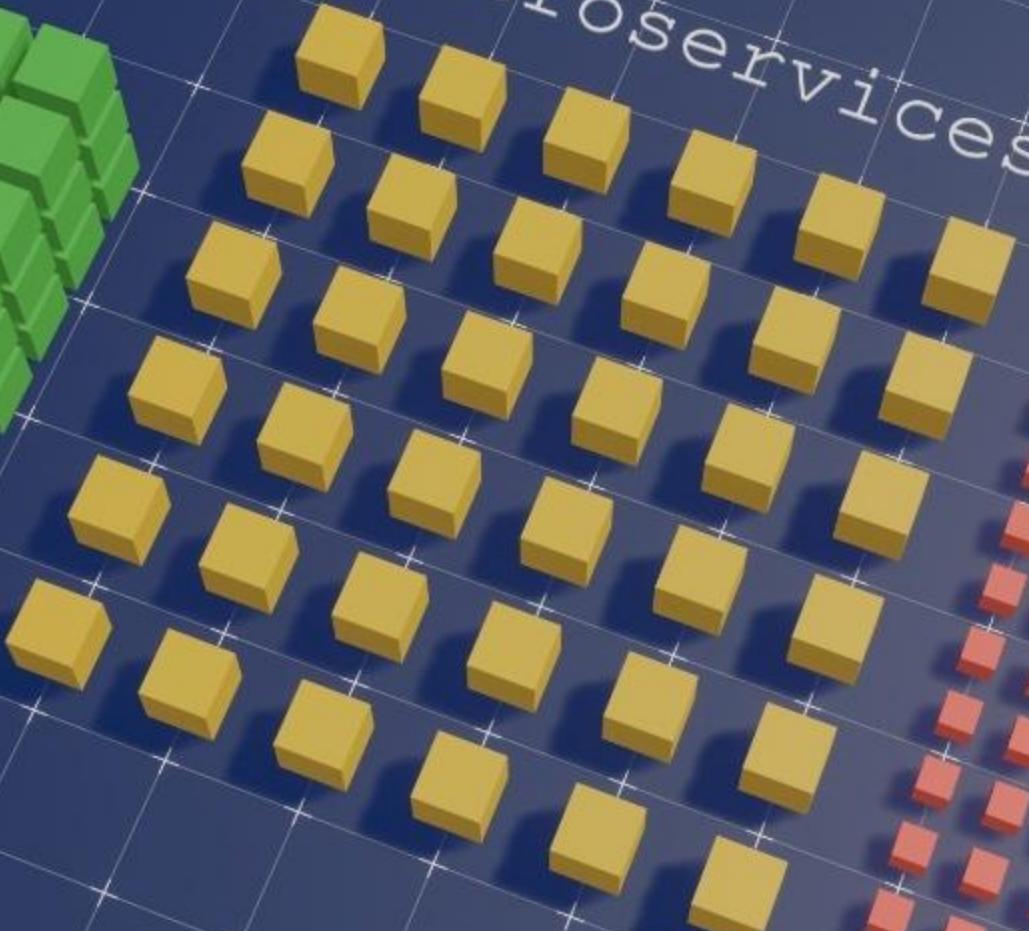


Monolith
Modules

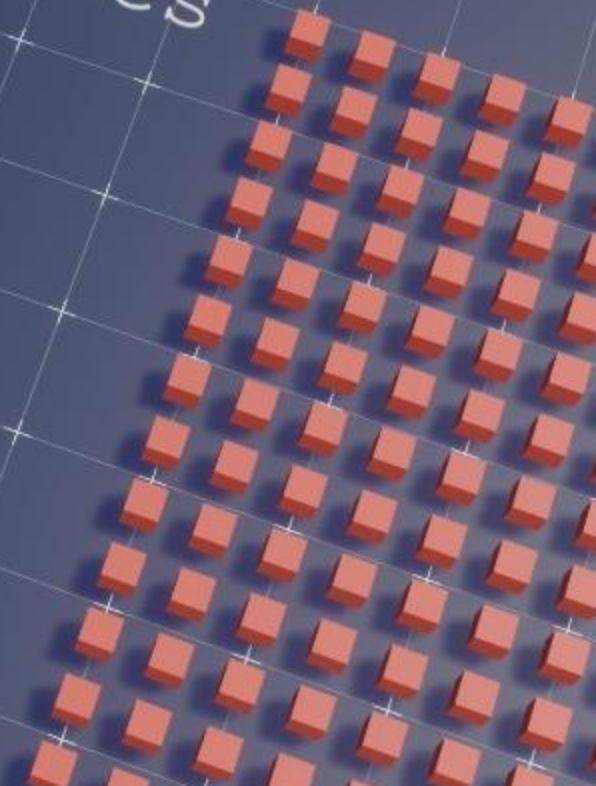
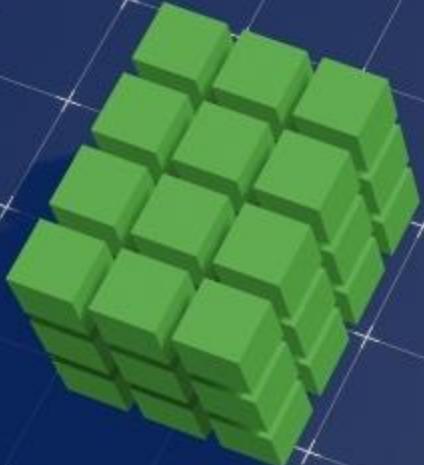
Microservices



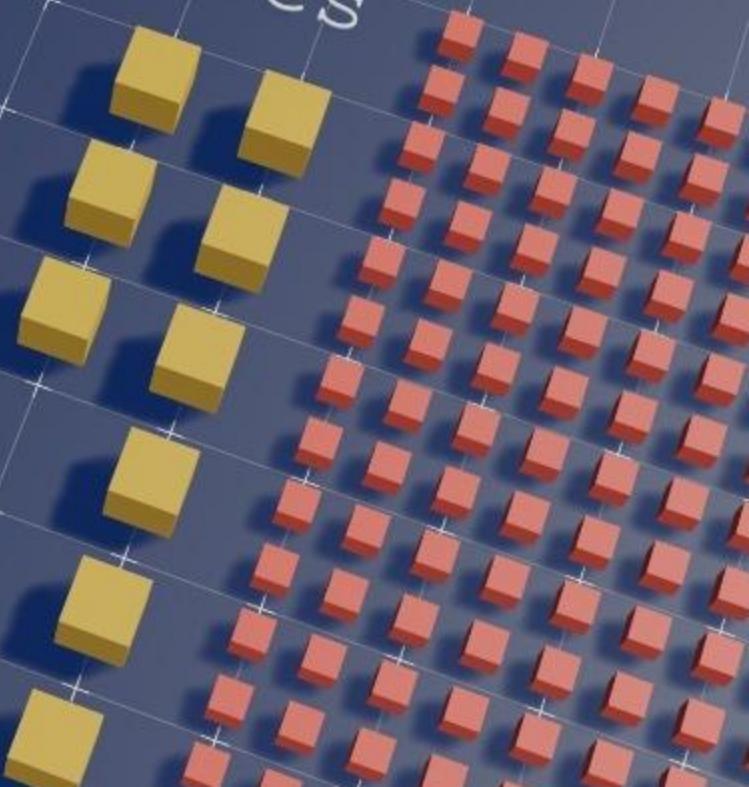
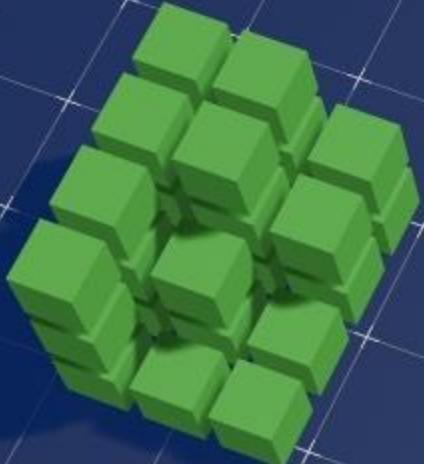
Microservices



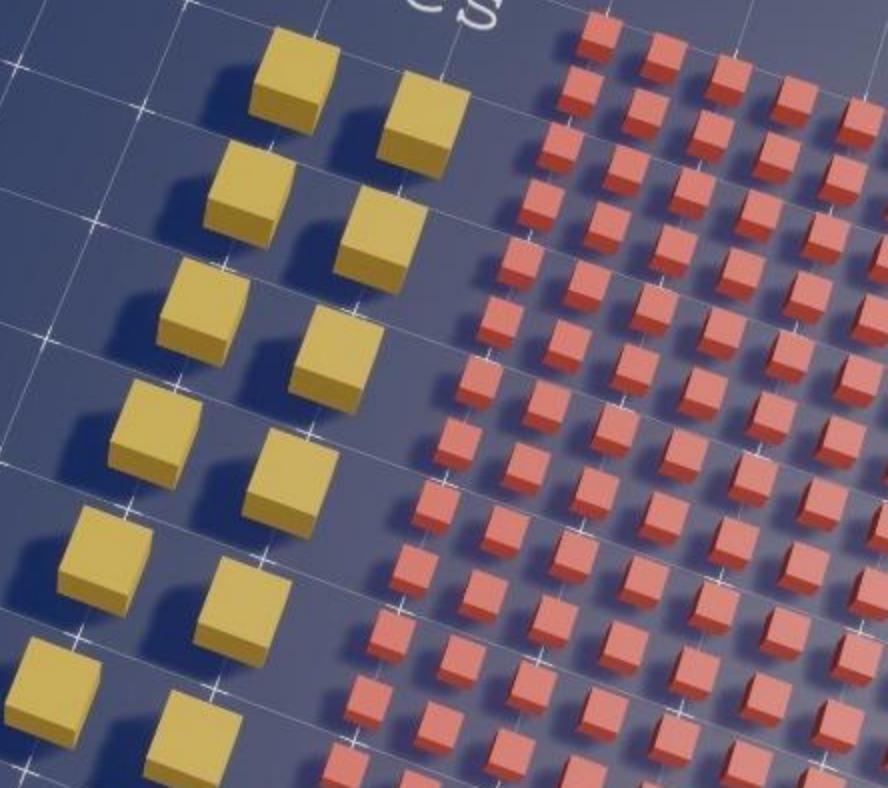
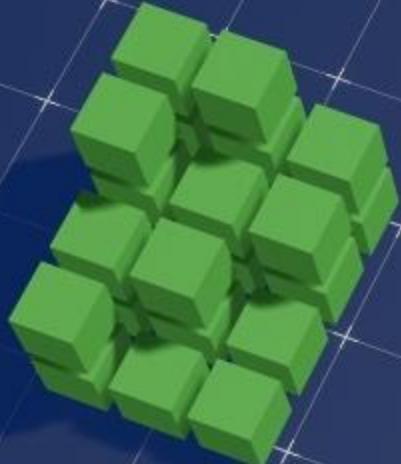
Microservices



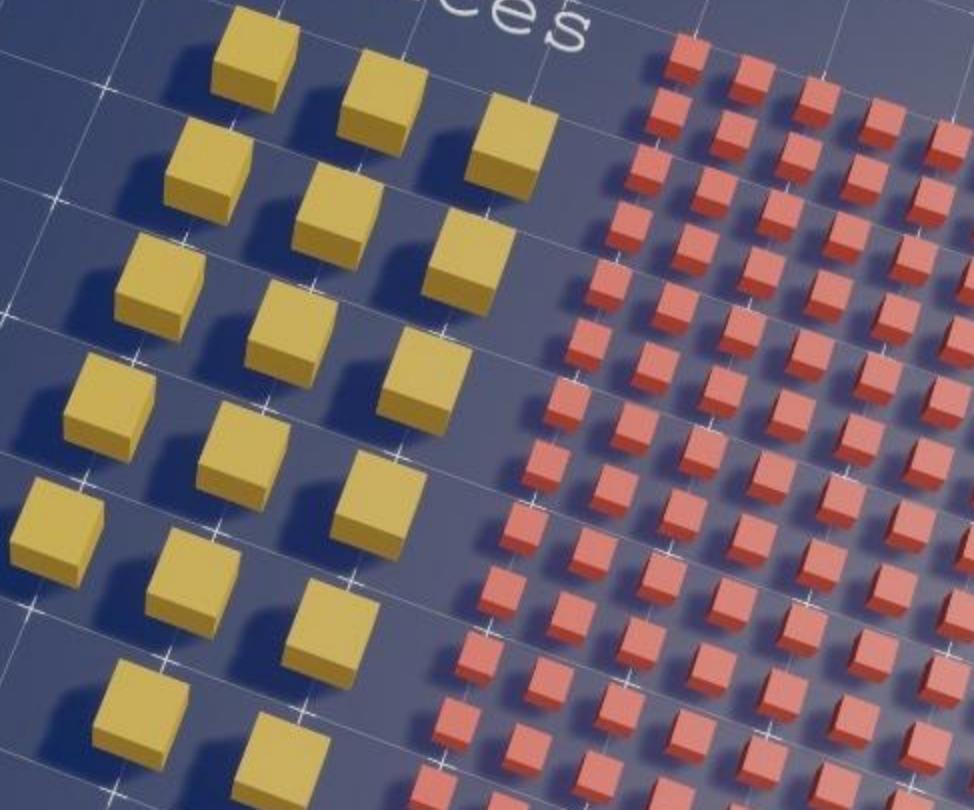
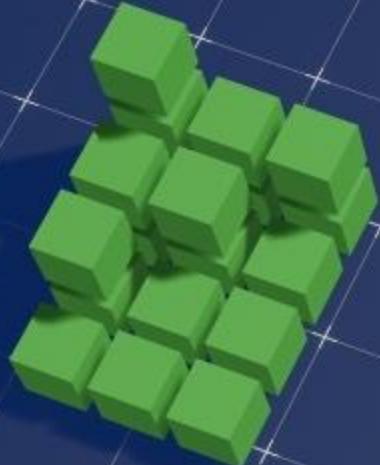
Microservices



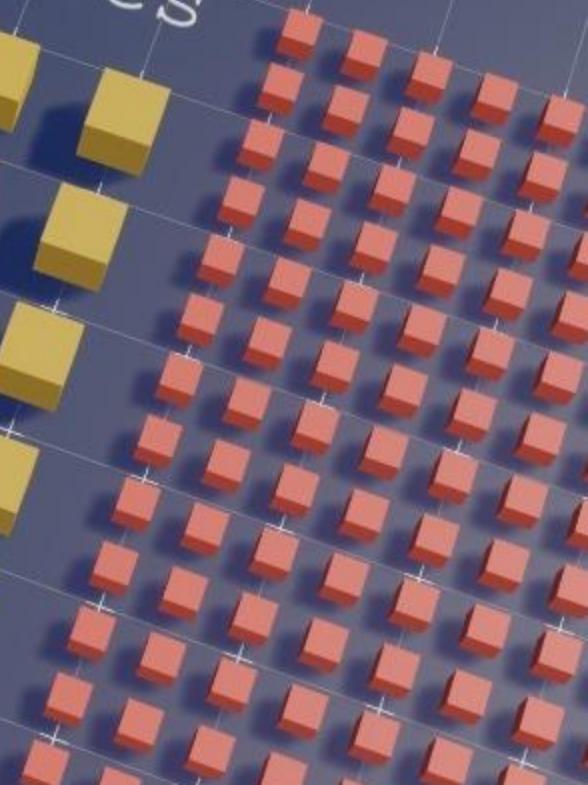
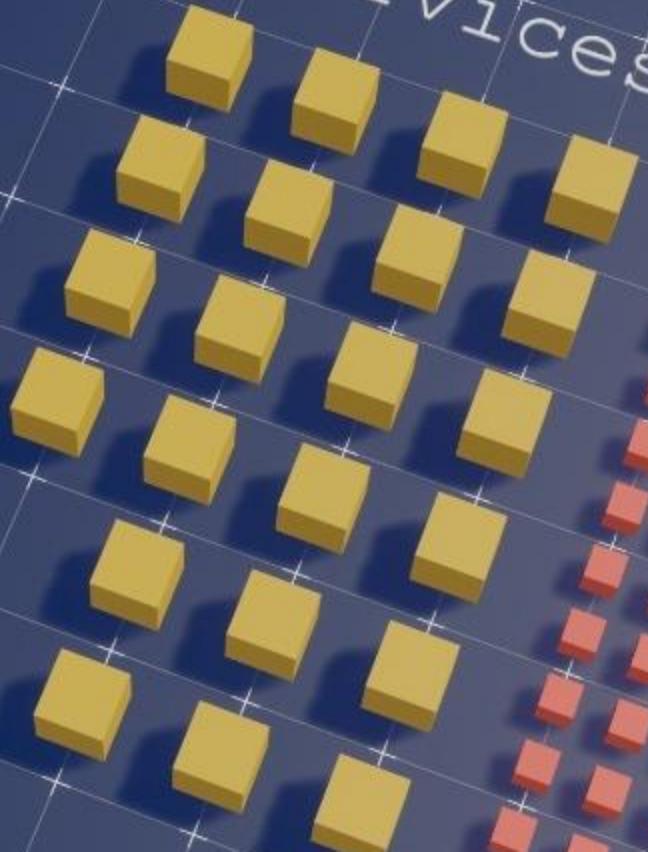
Microservices



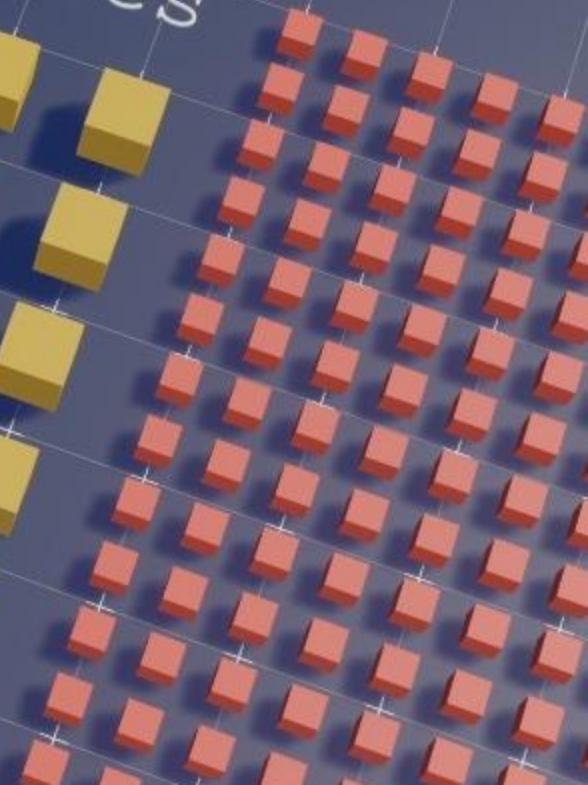
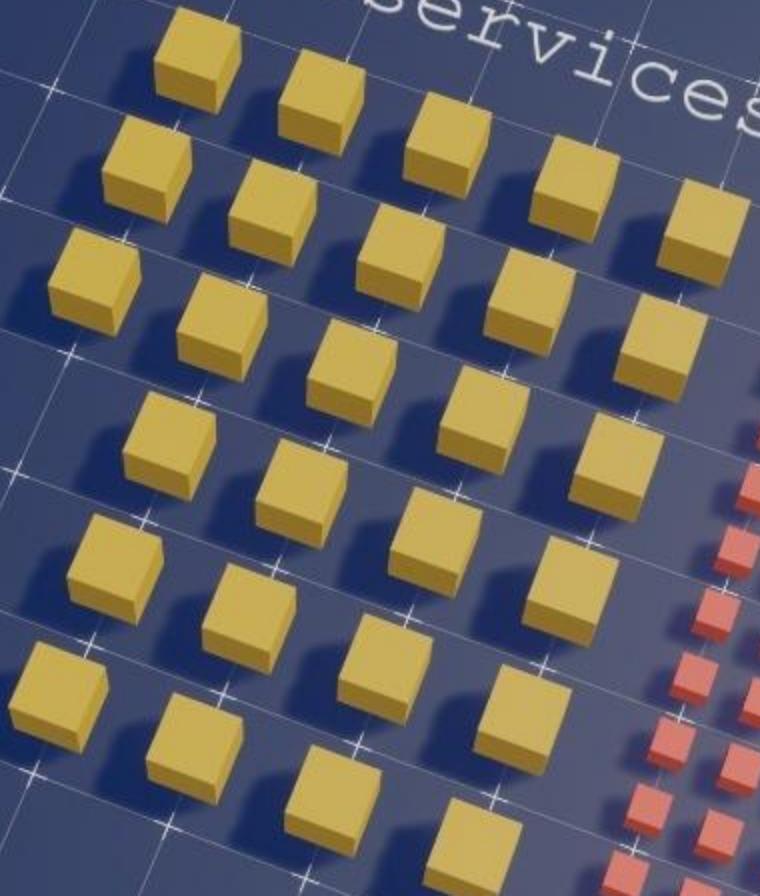
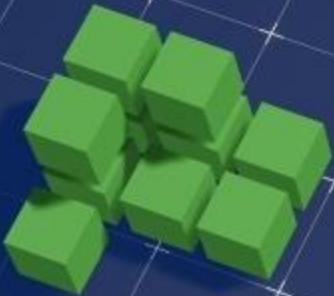
Microservices



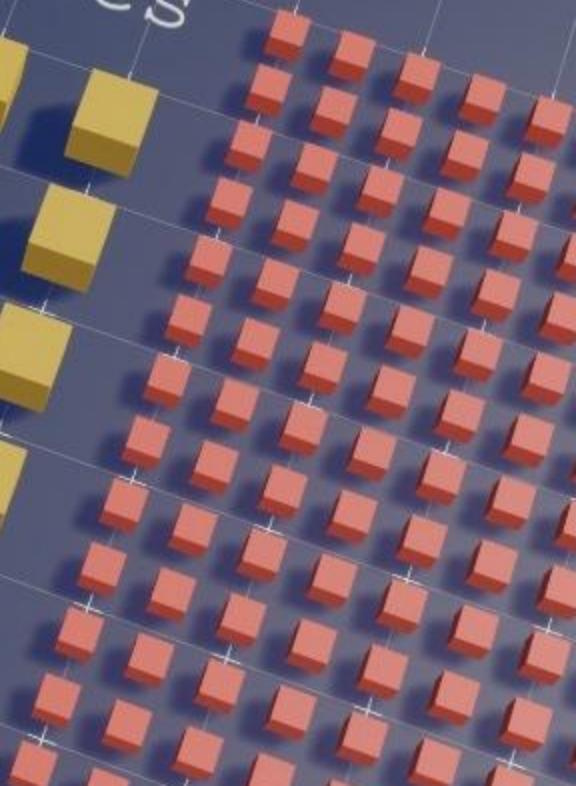
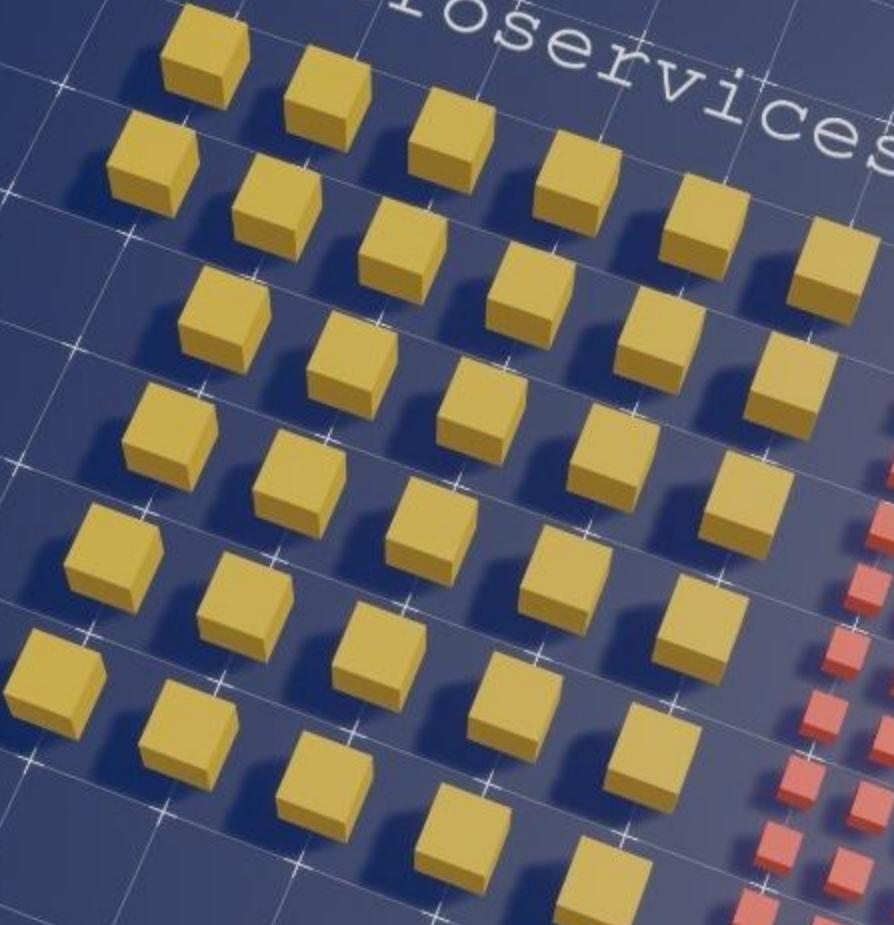
Microservices



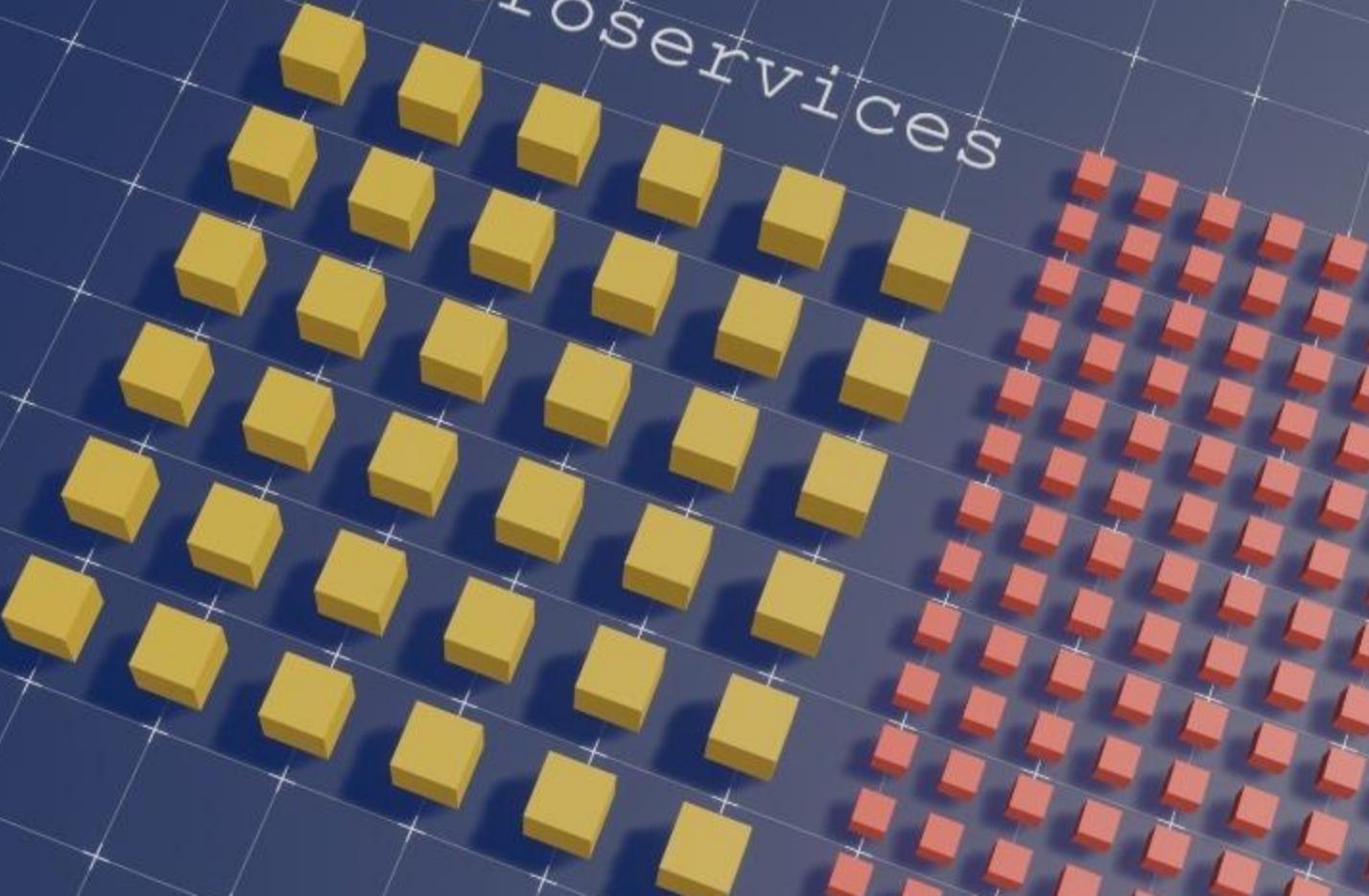
Microservices

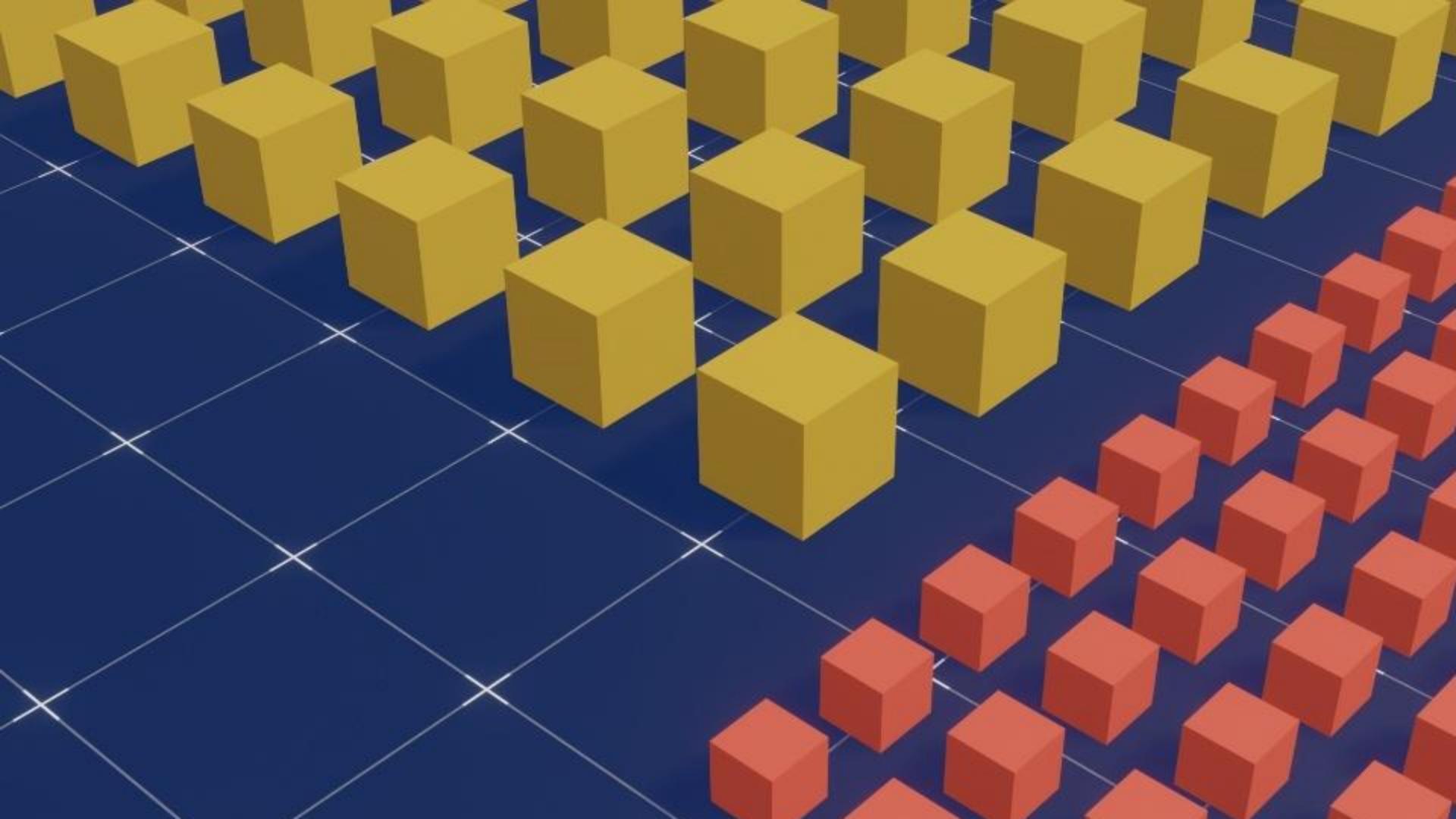


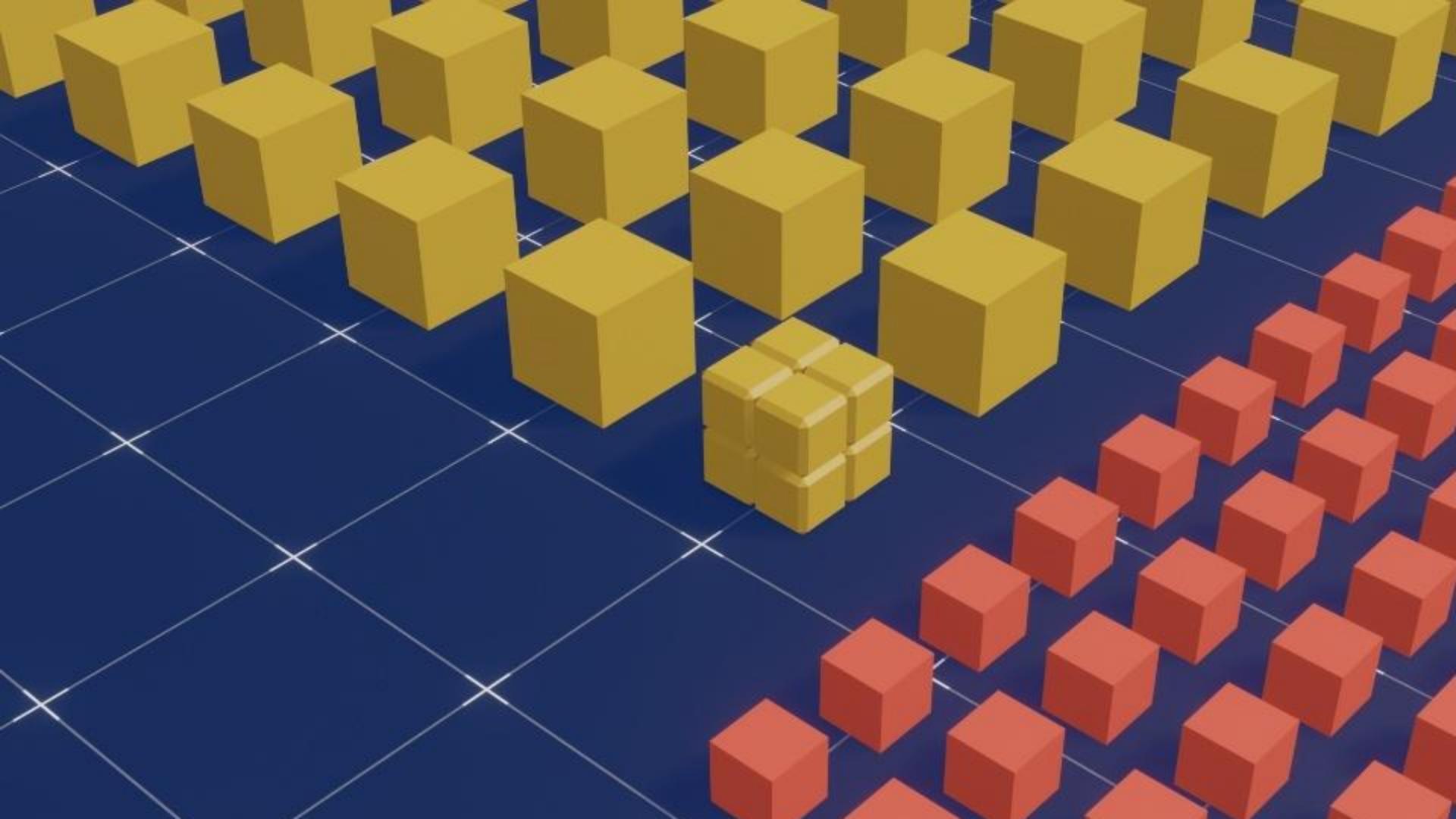
Microservices

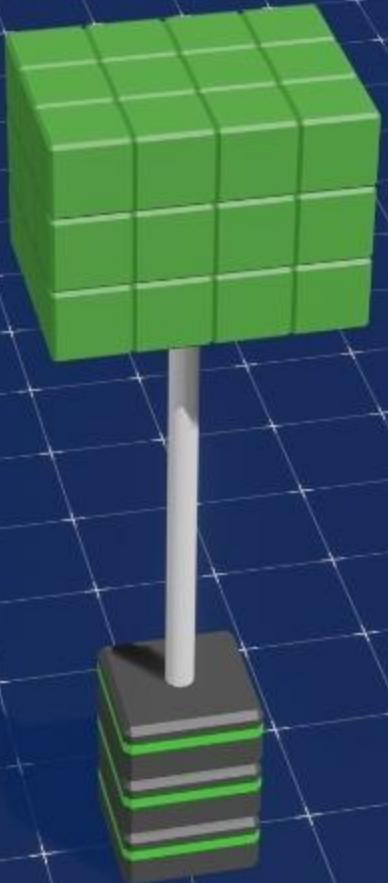


Microservices

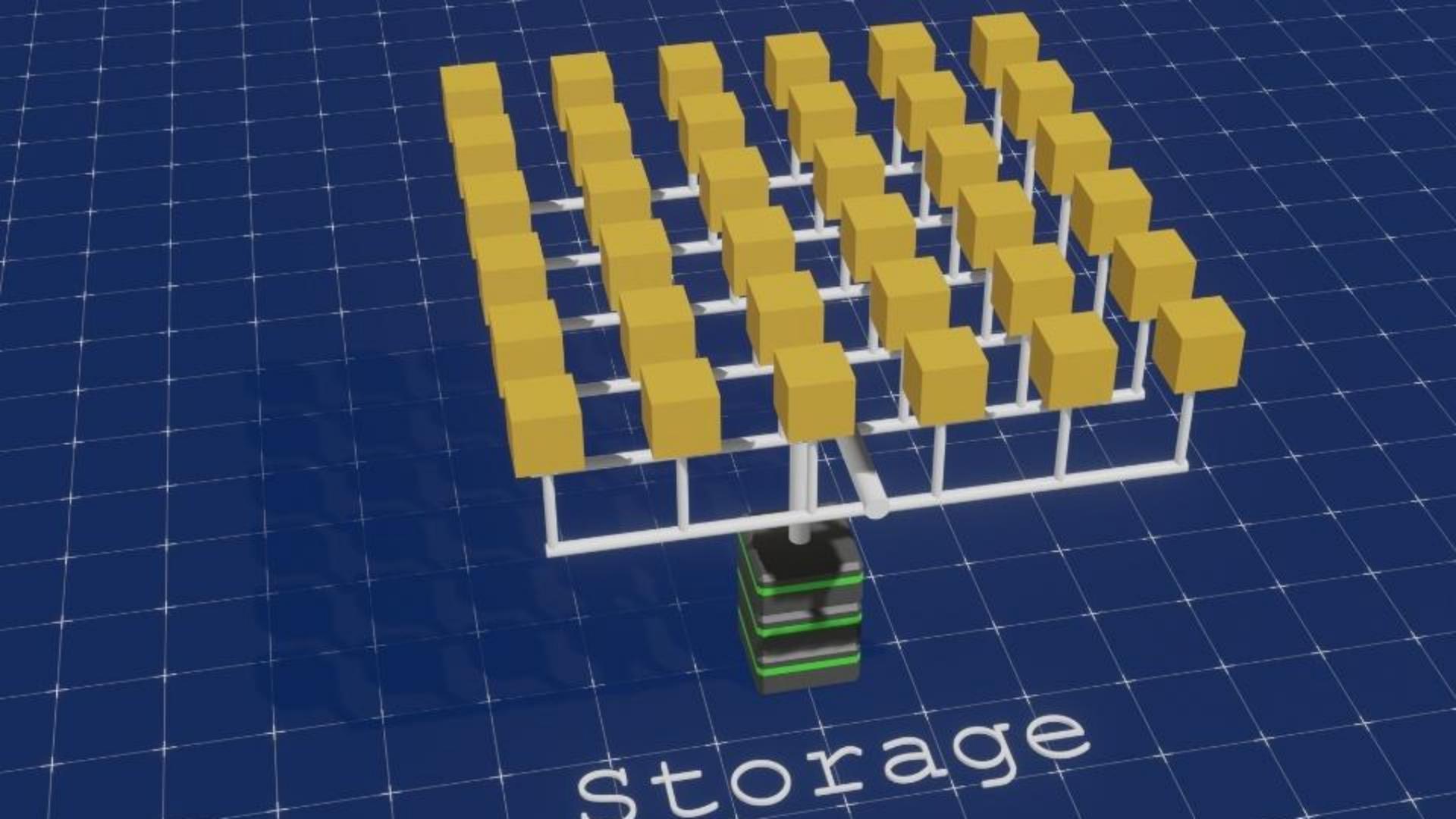




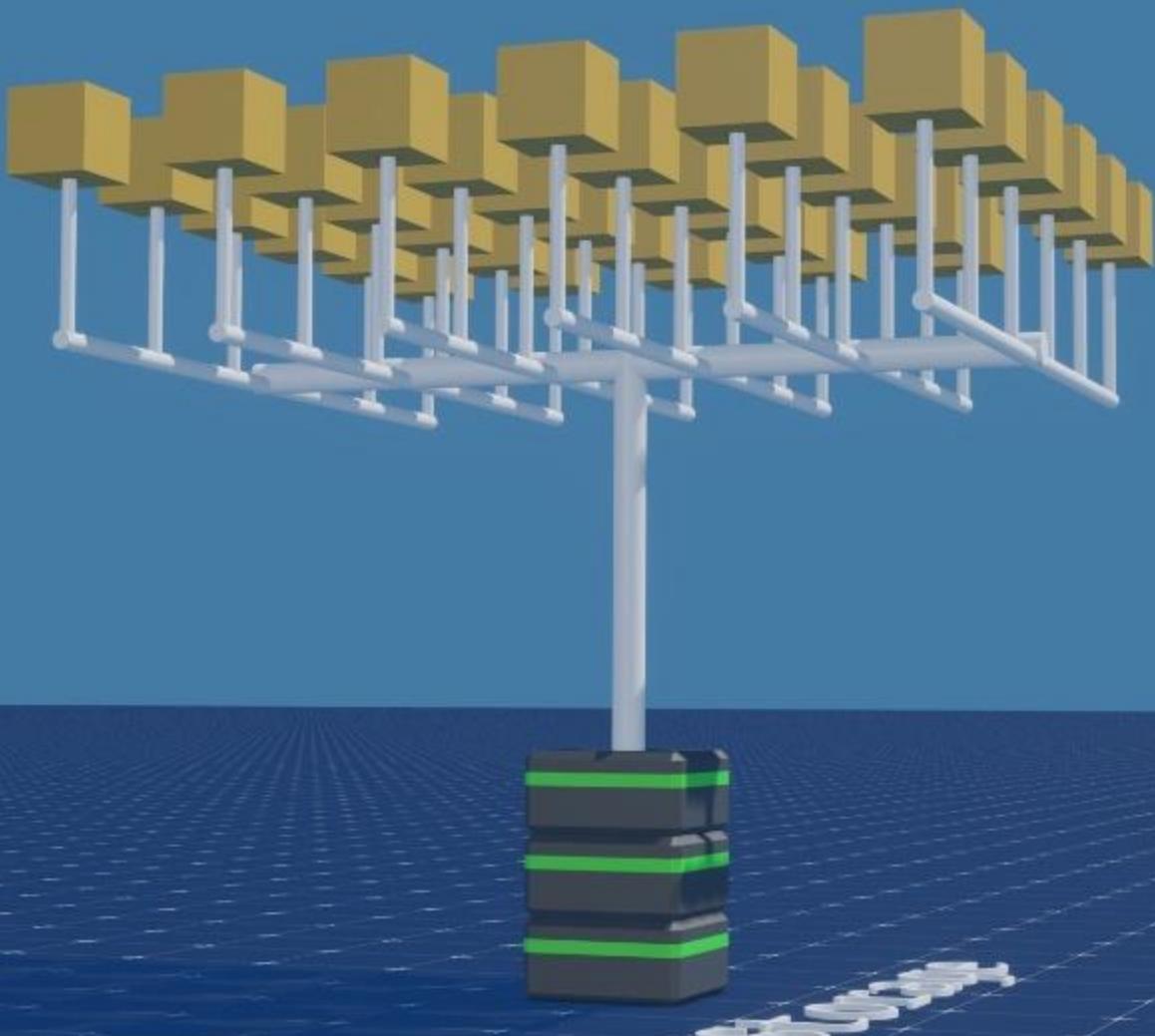


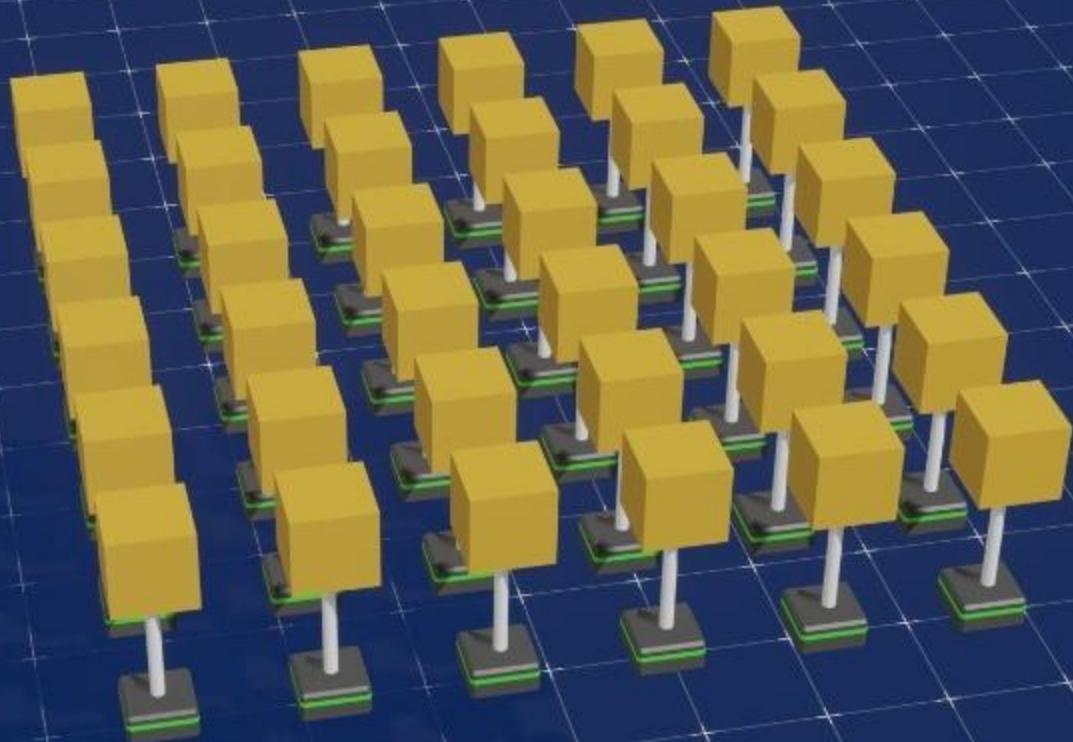


Storage

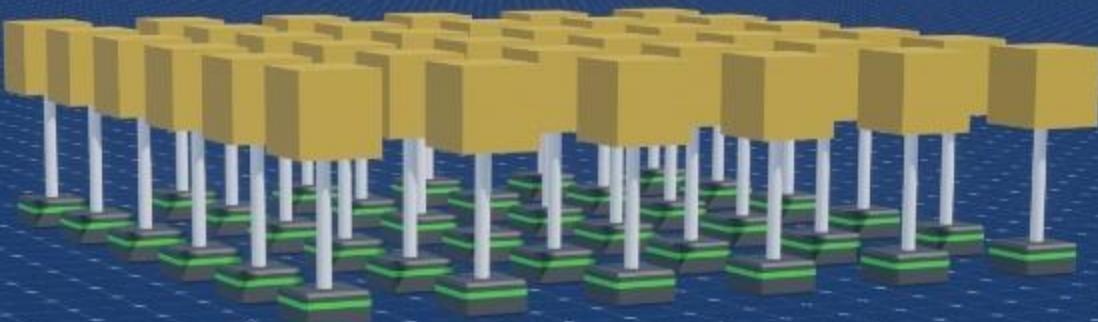


Storage

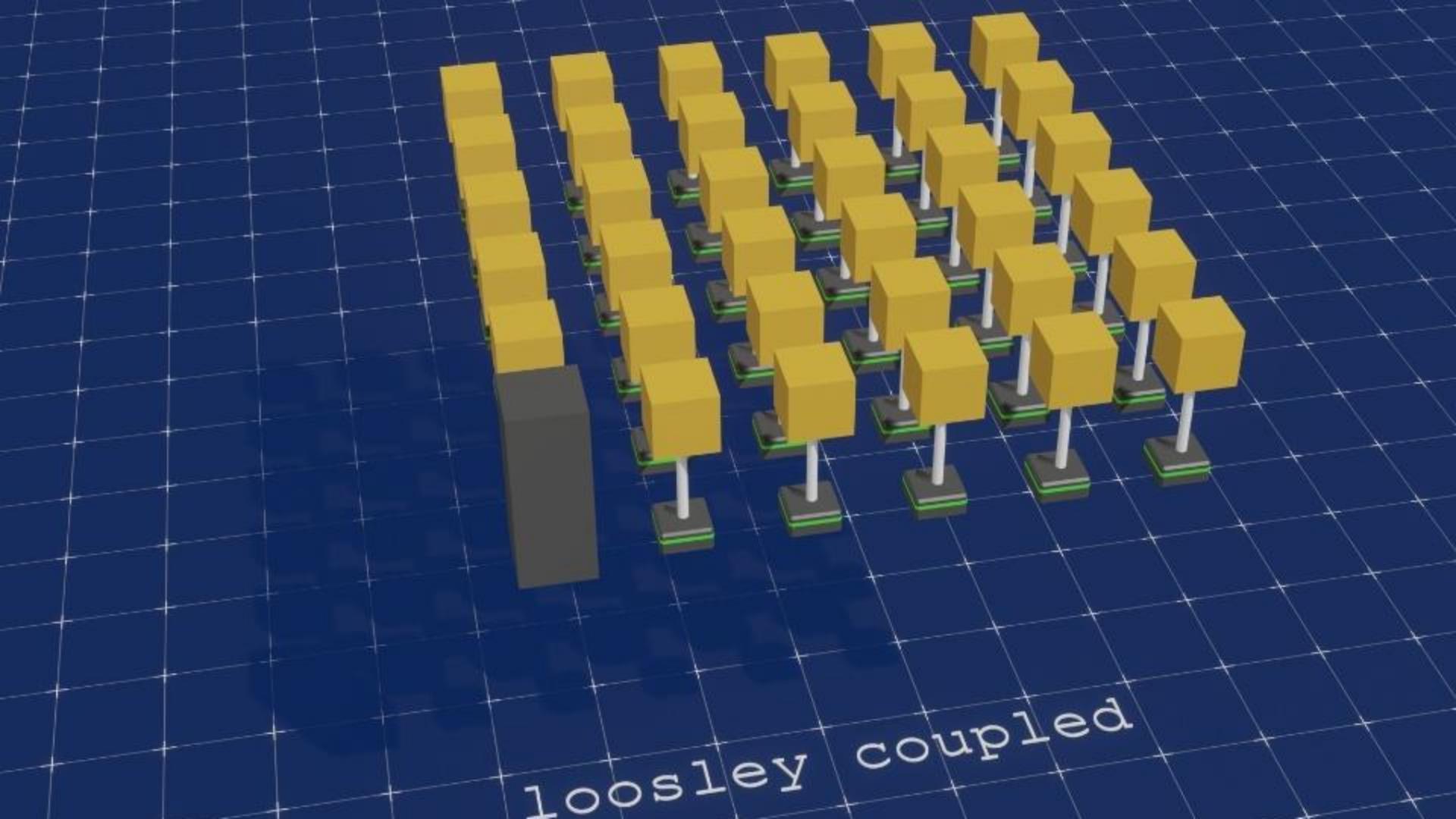




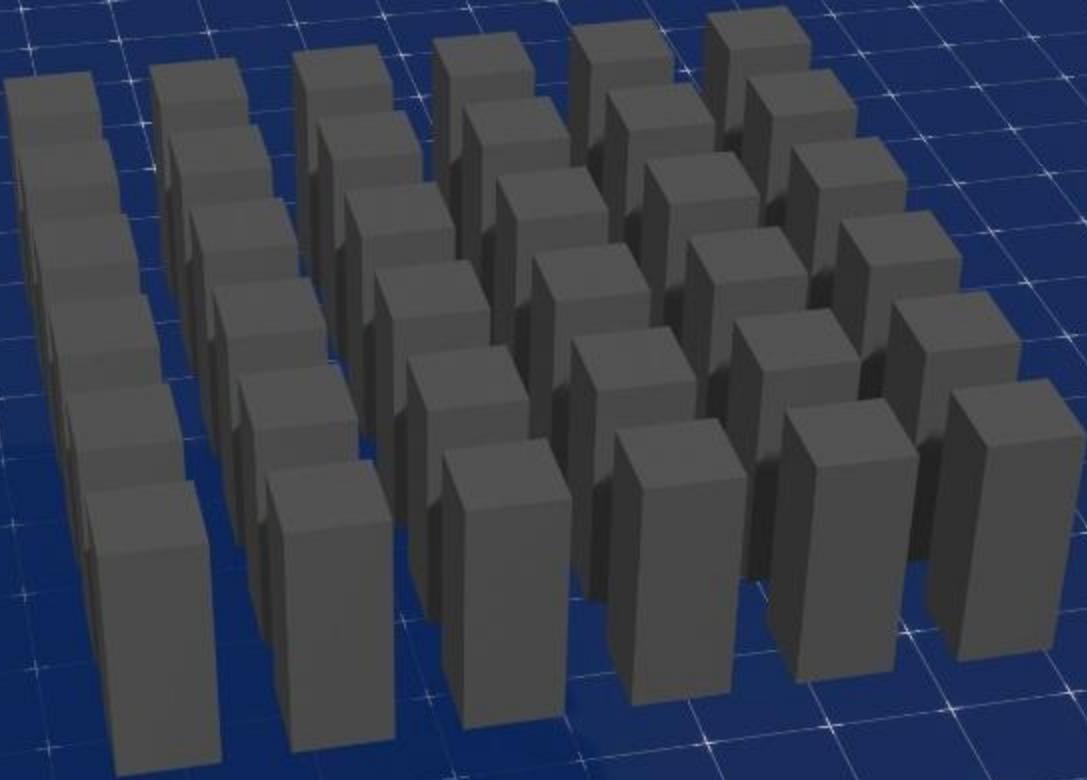
loosley coupled



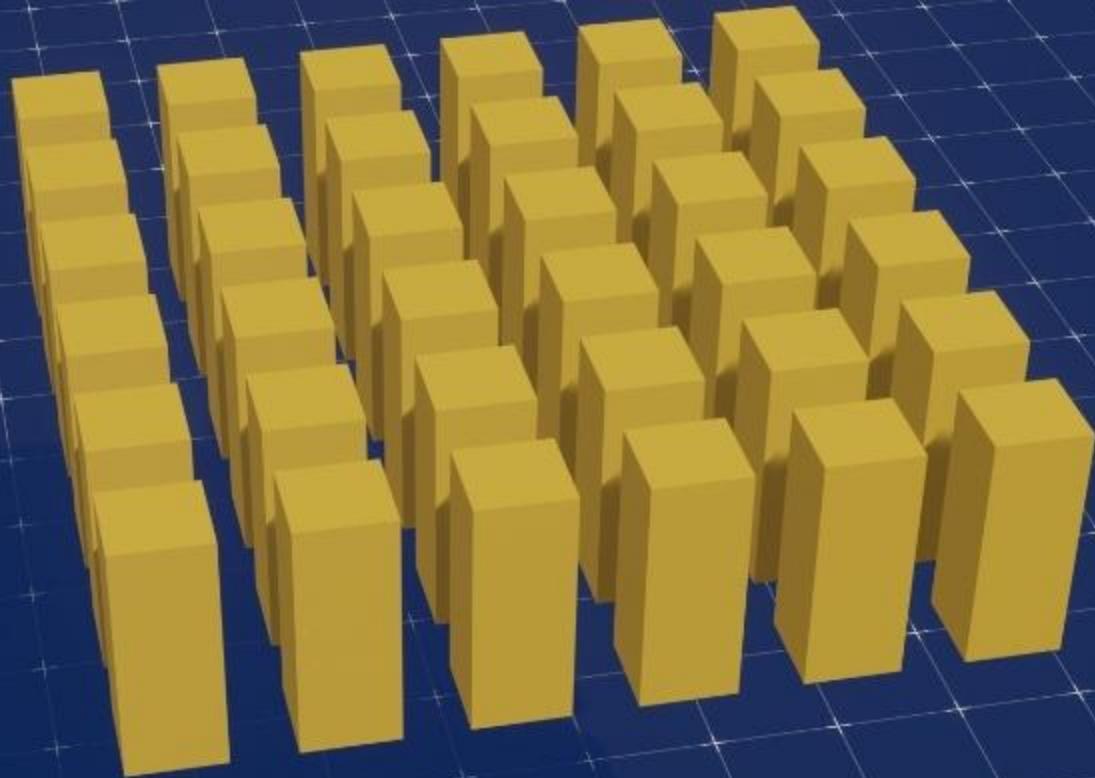
loosely coupled



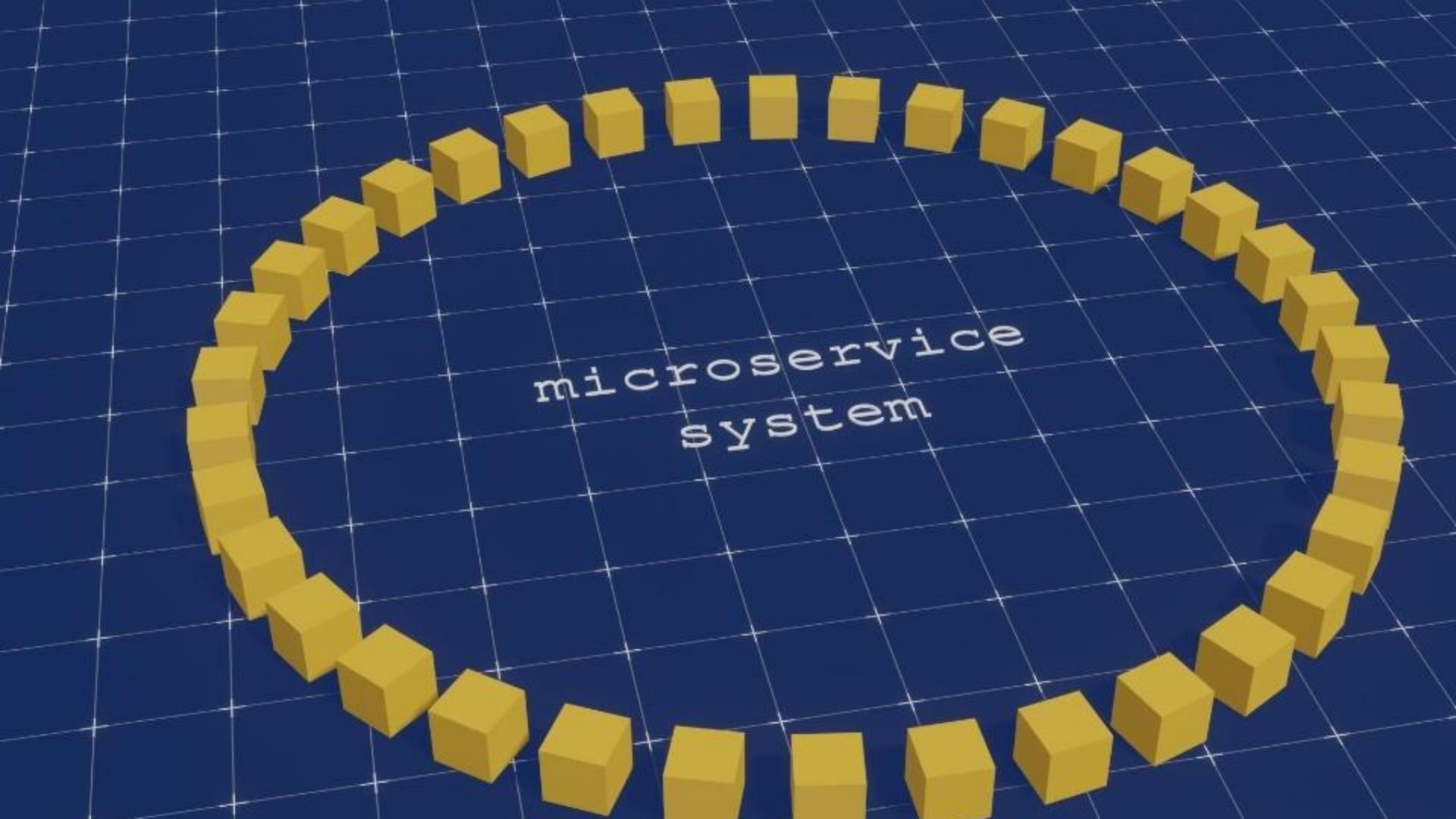
loosley coupled



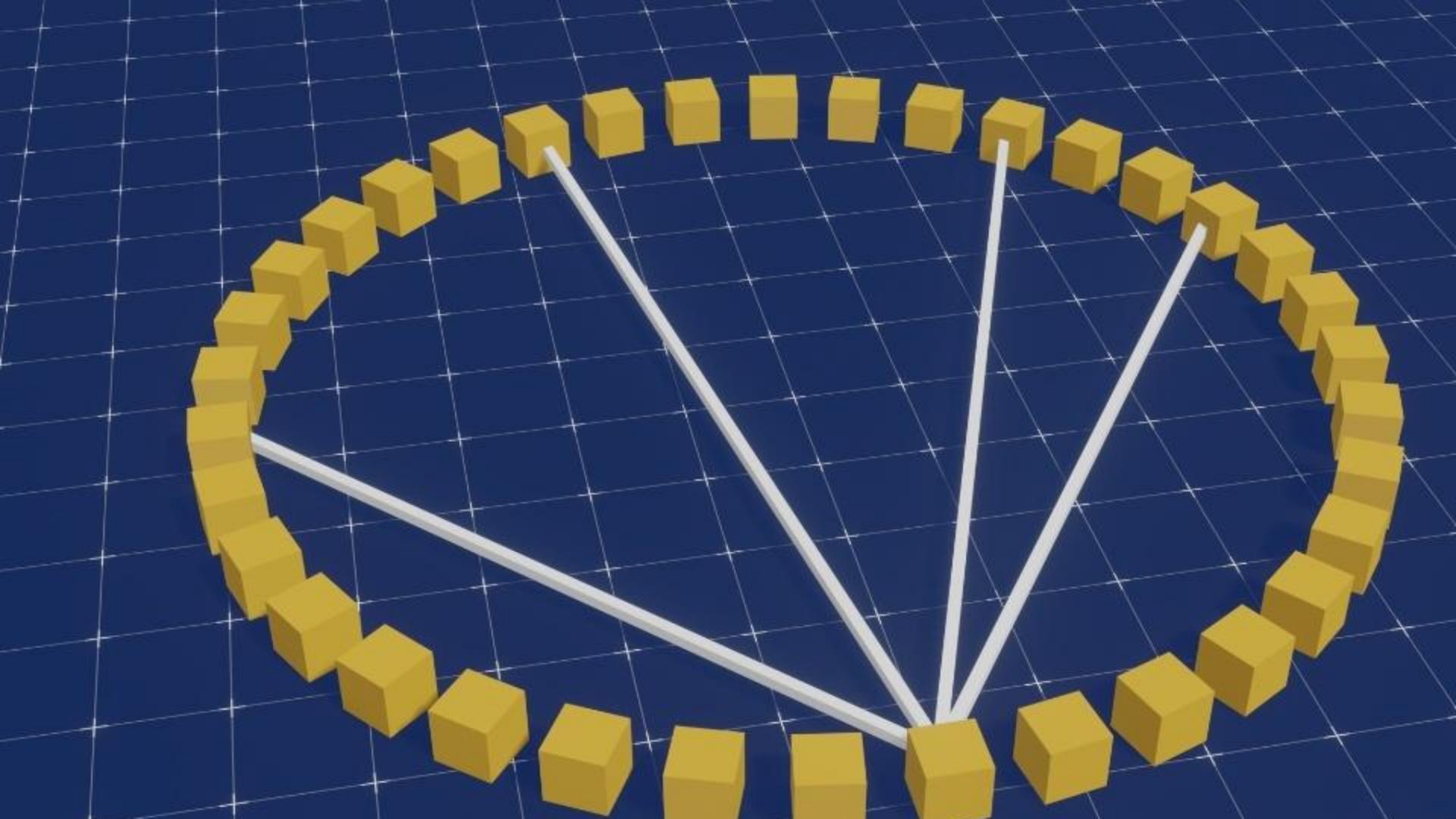
loosley coupled

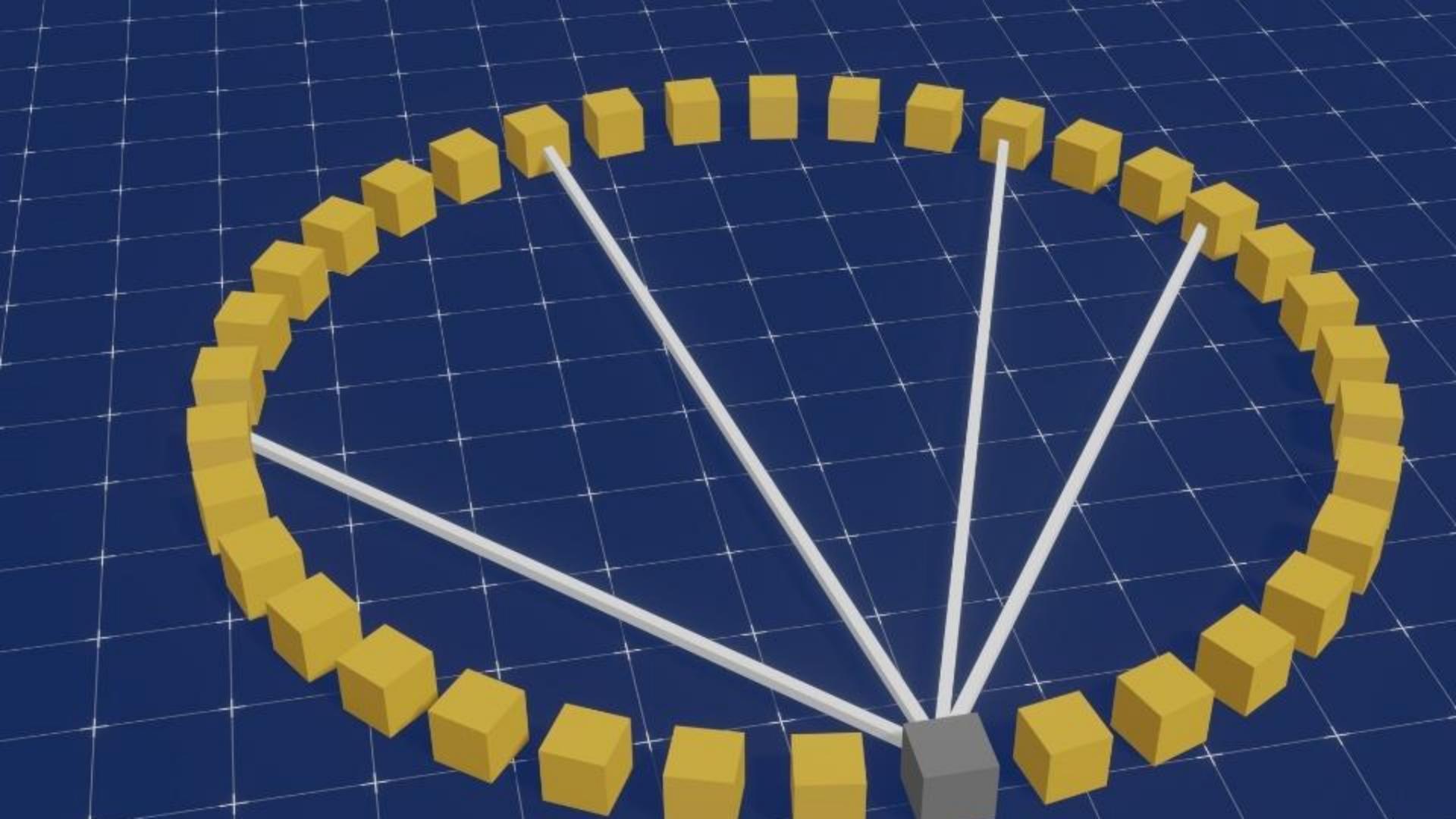


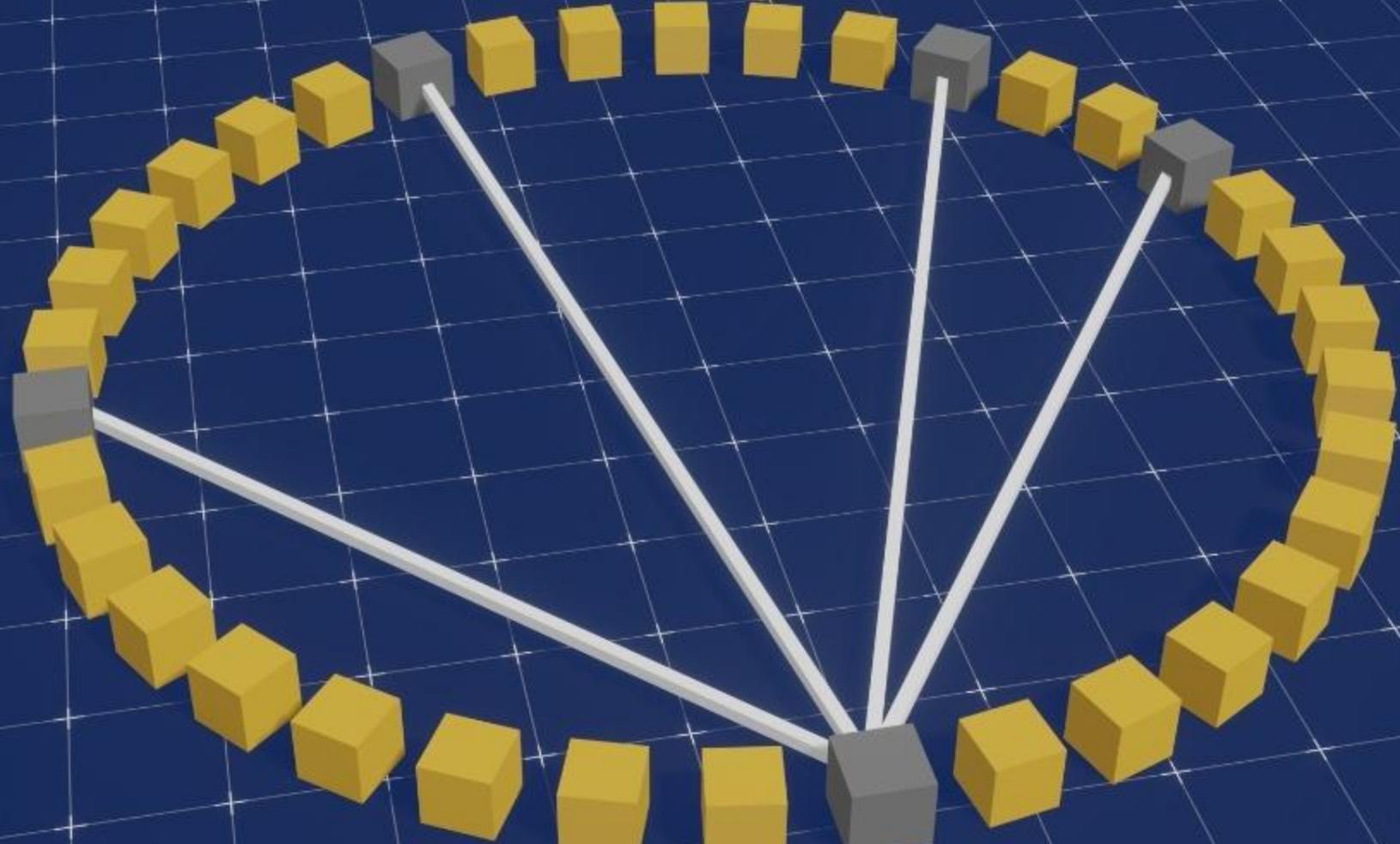
loosley coupled



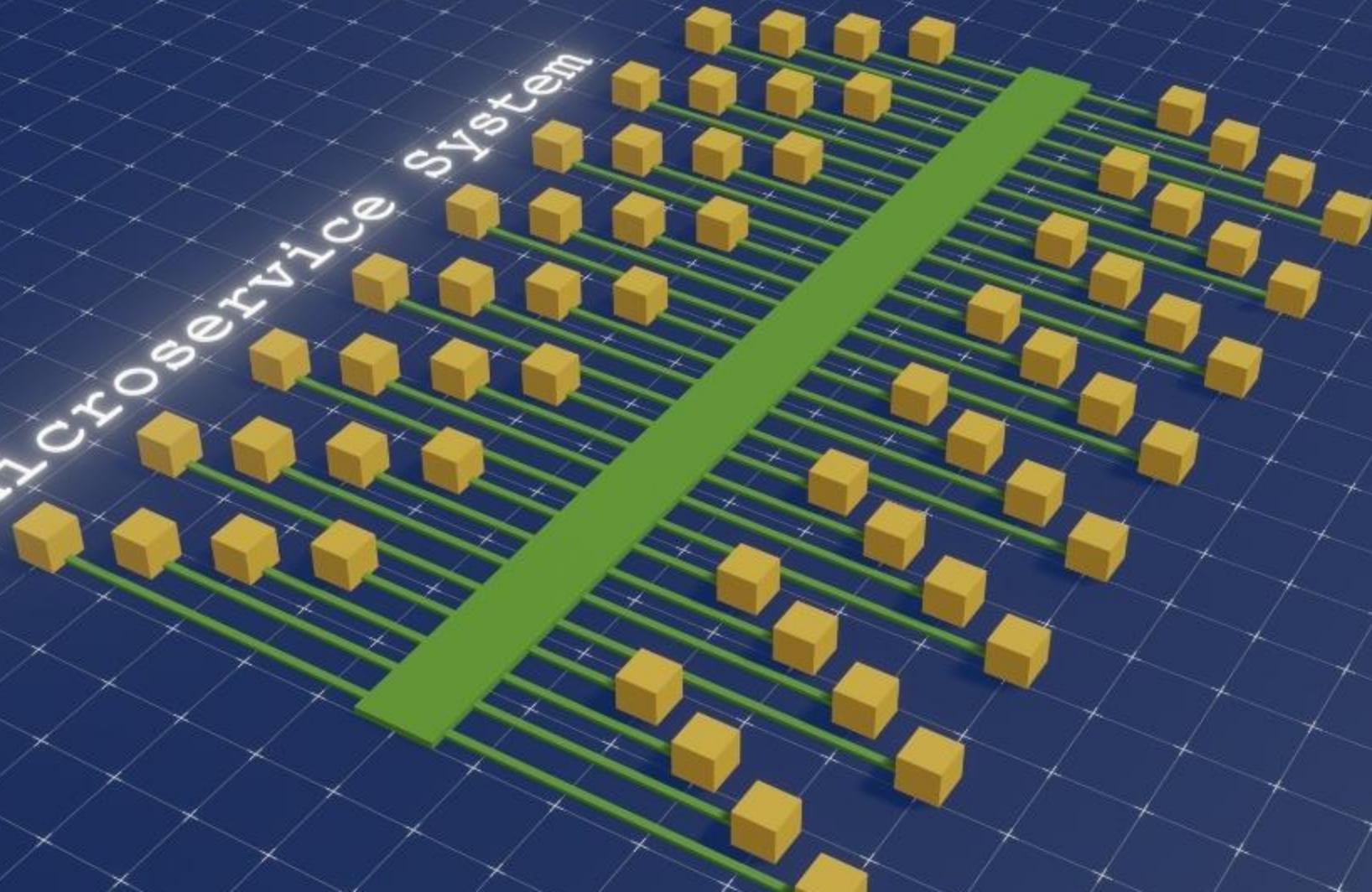
microservice
system

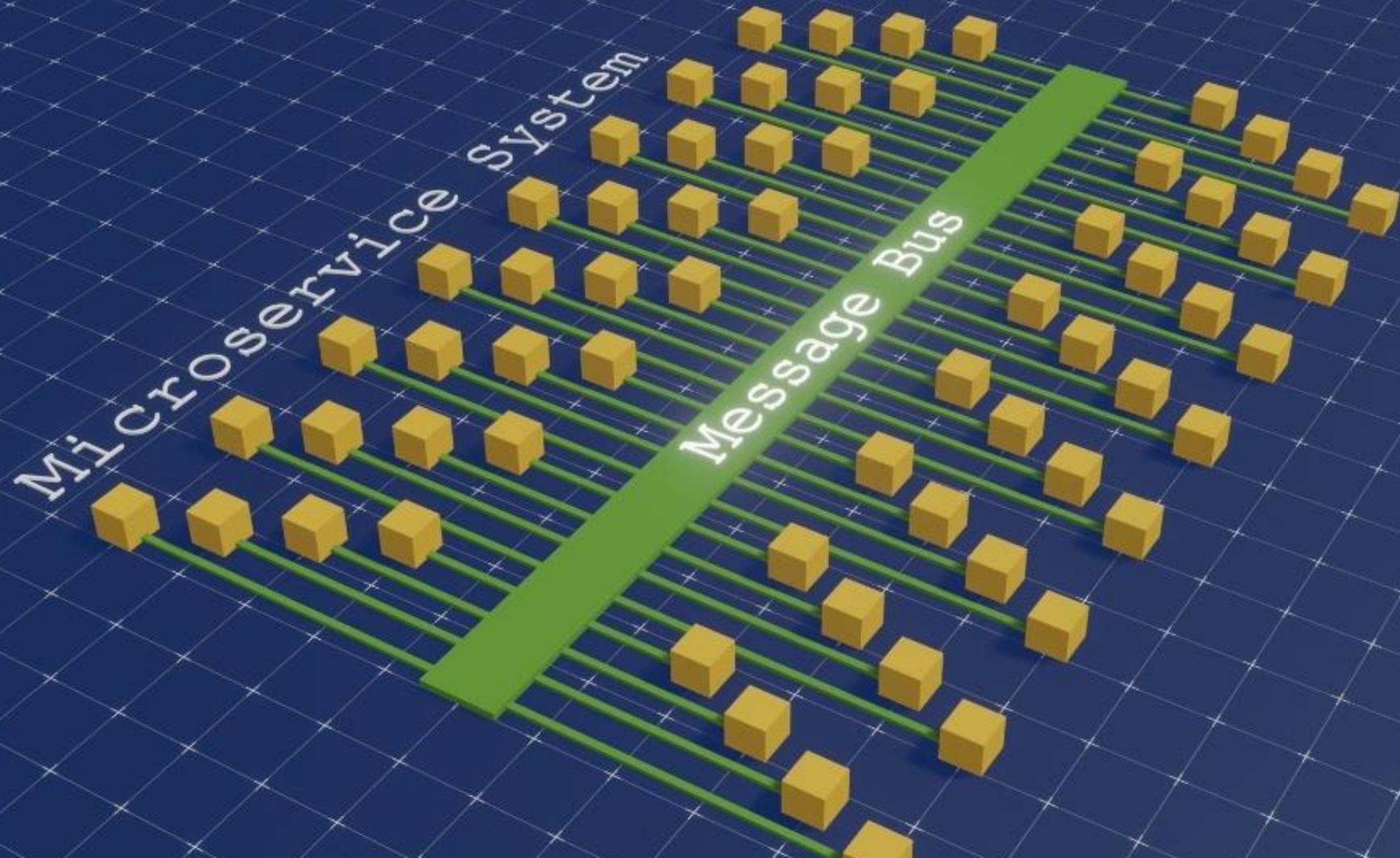


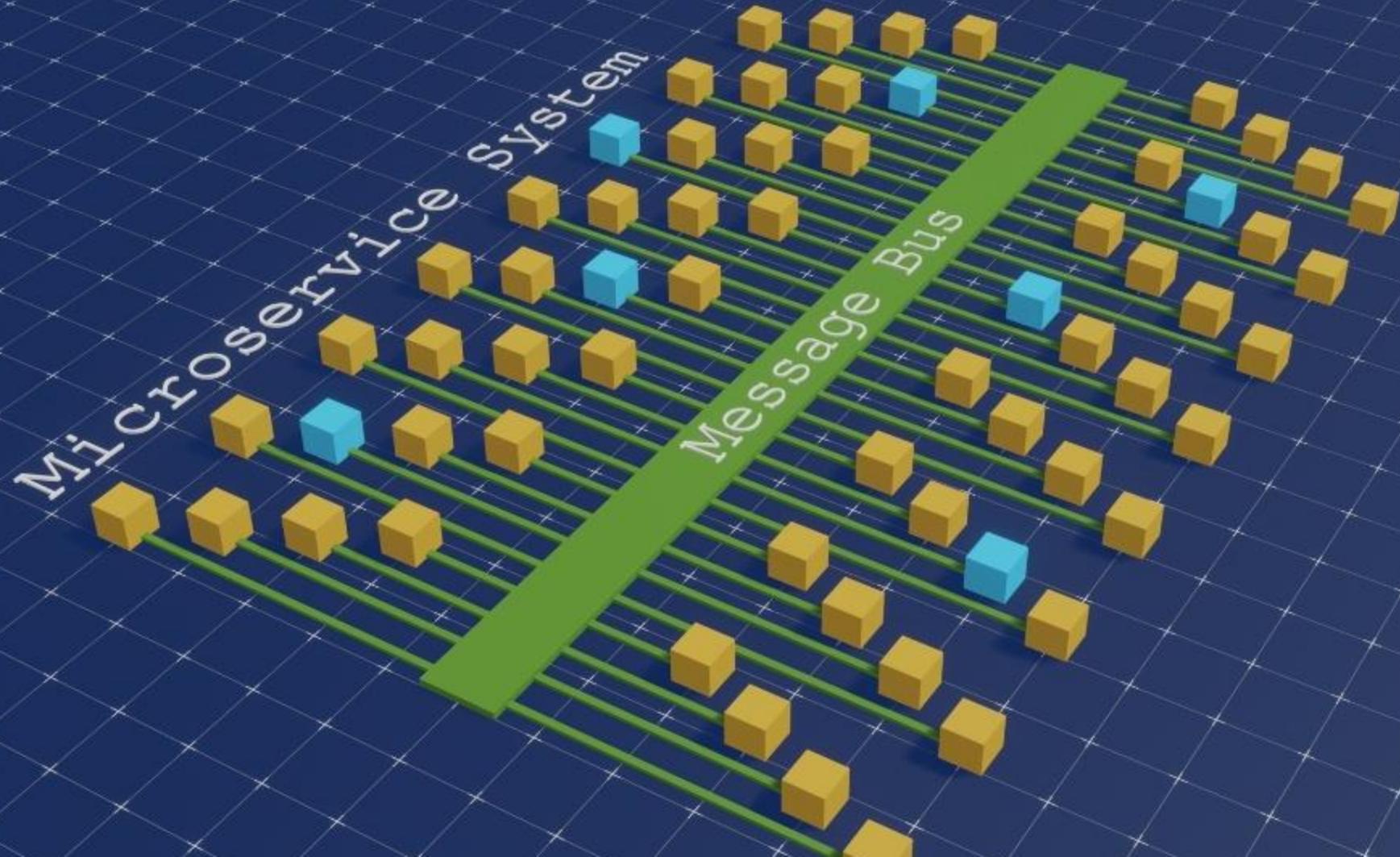


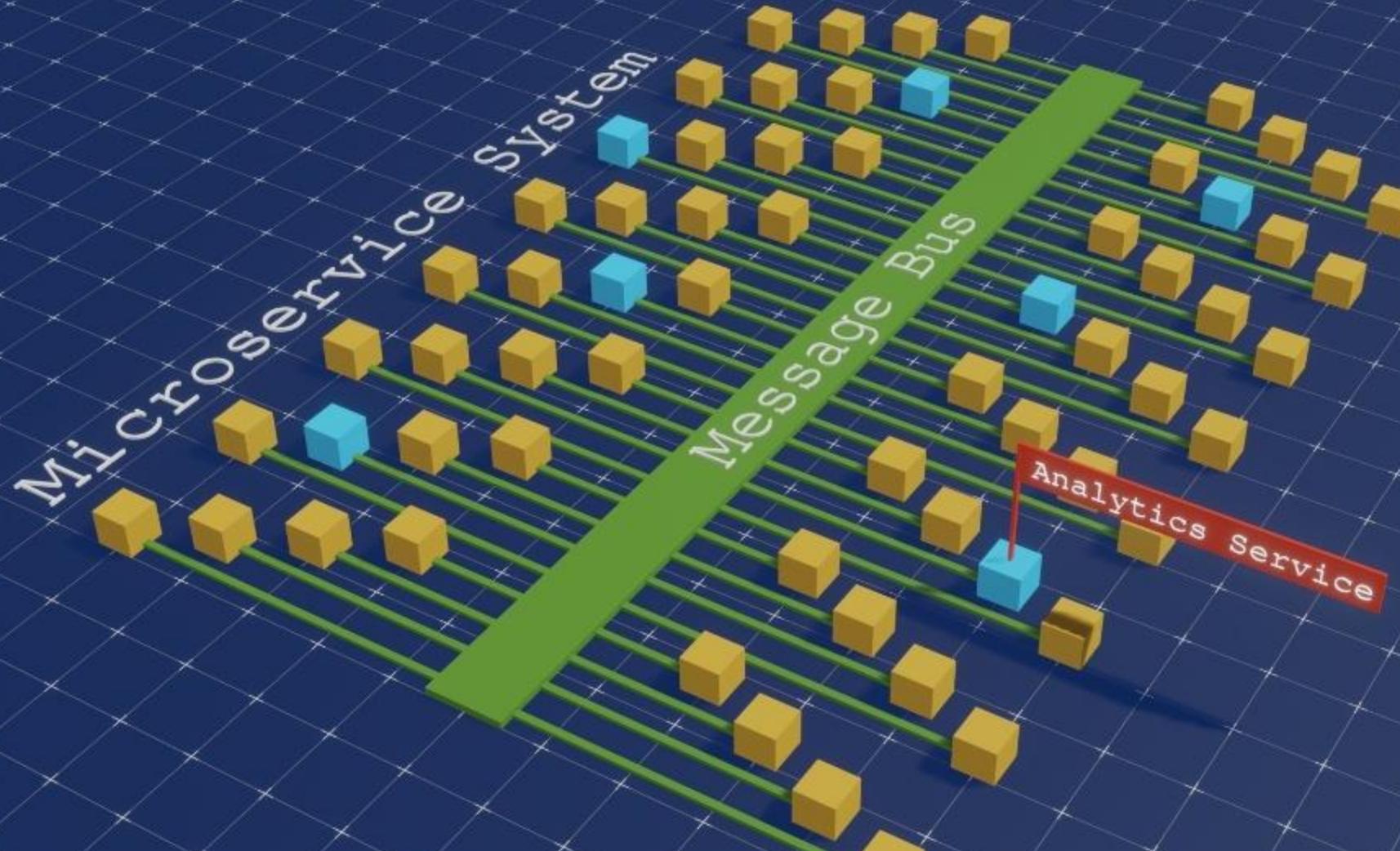


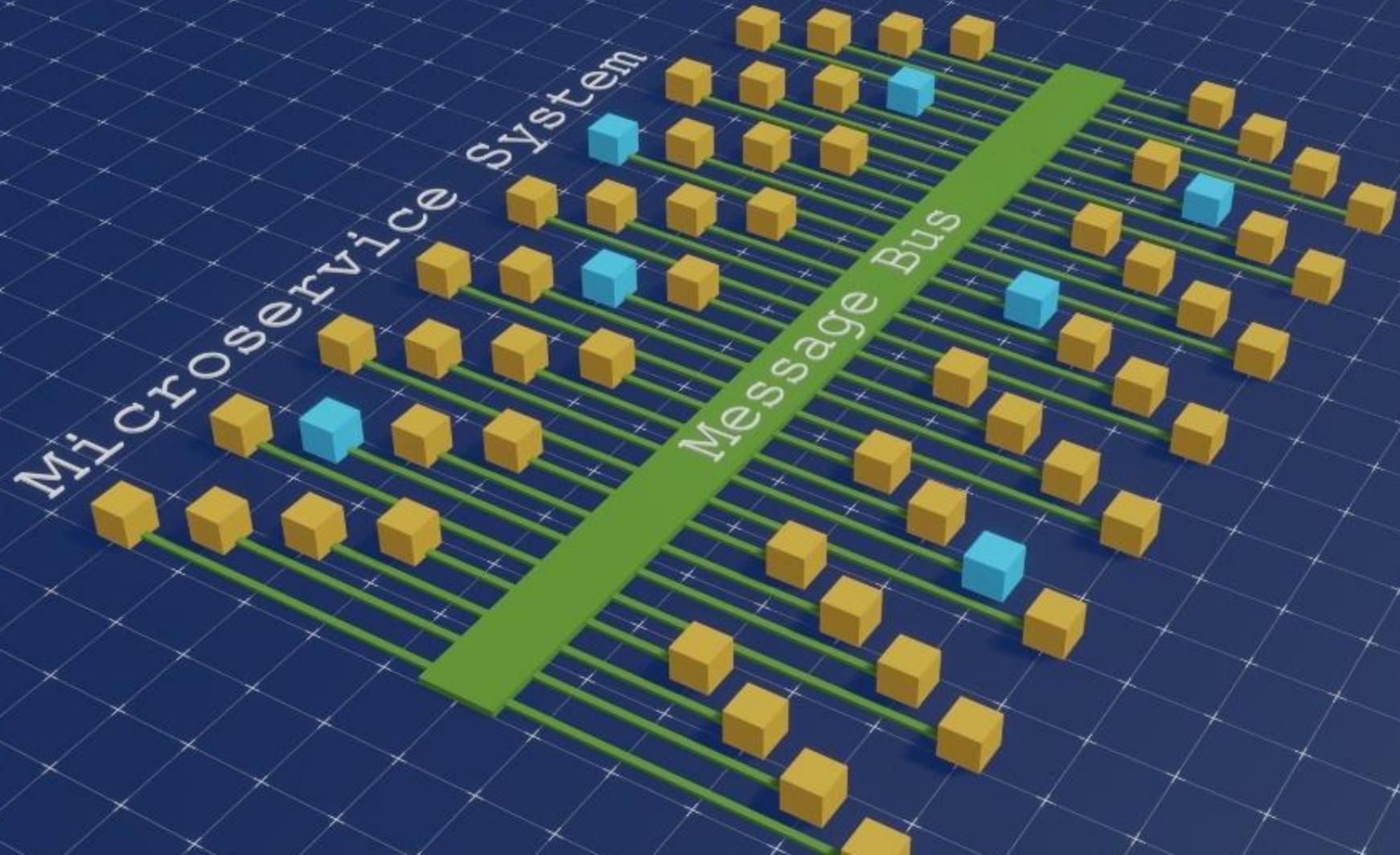
Microservice System

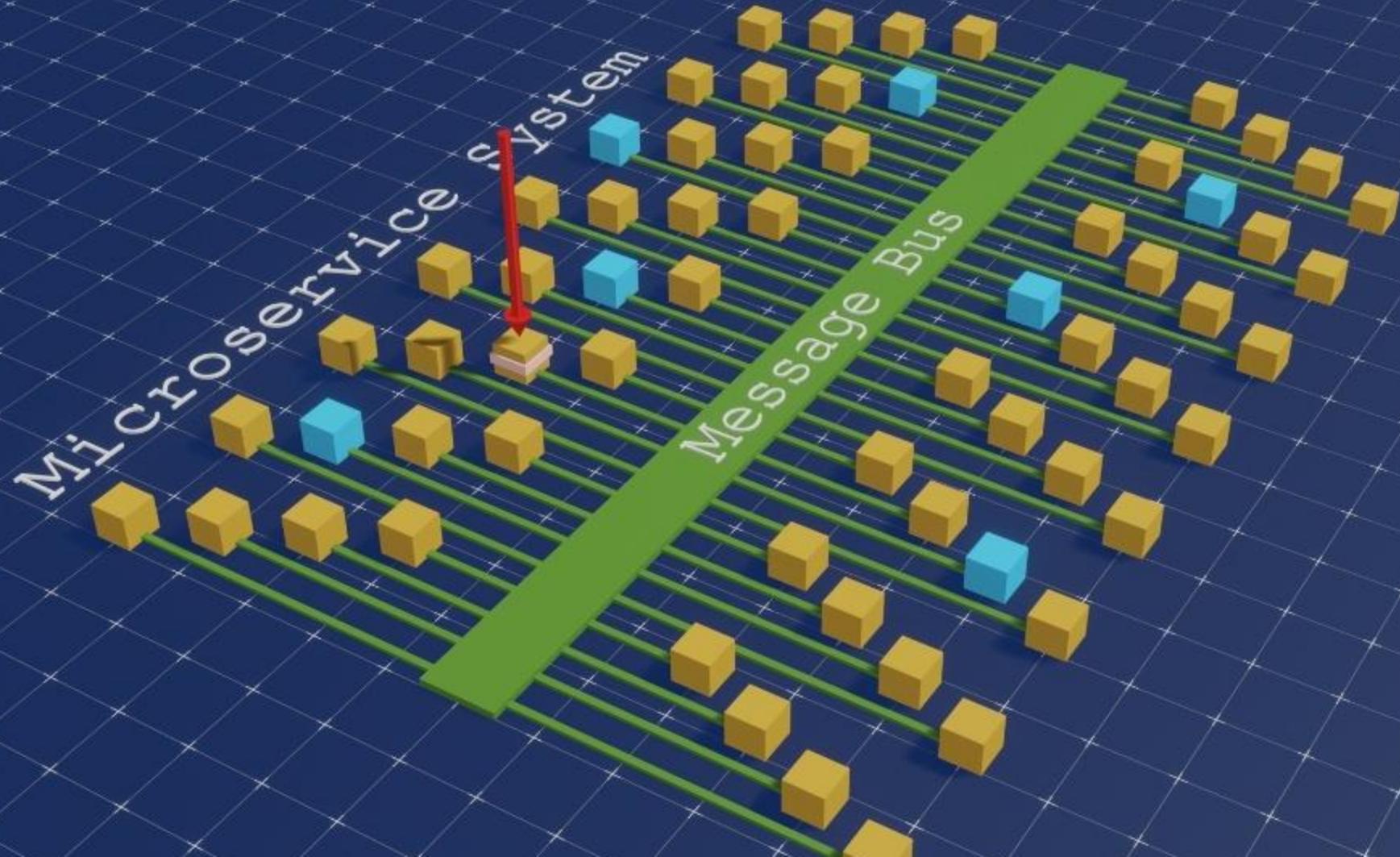


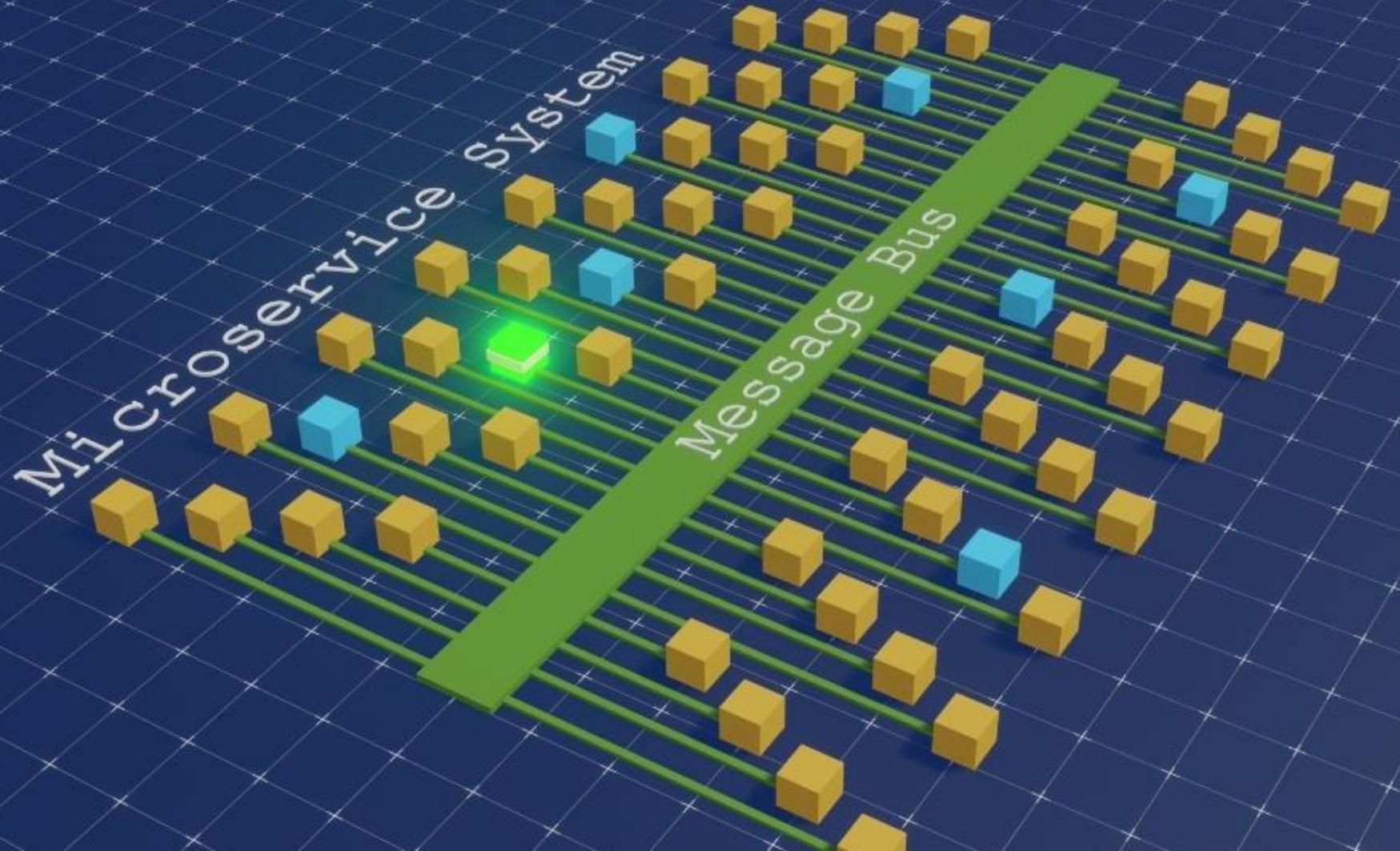


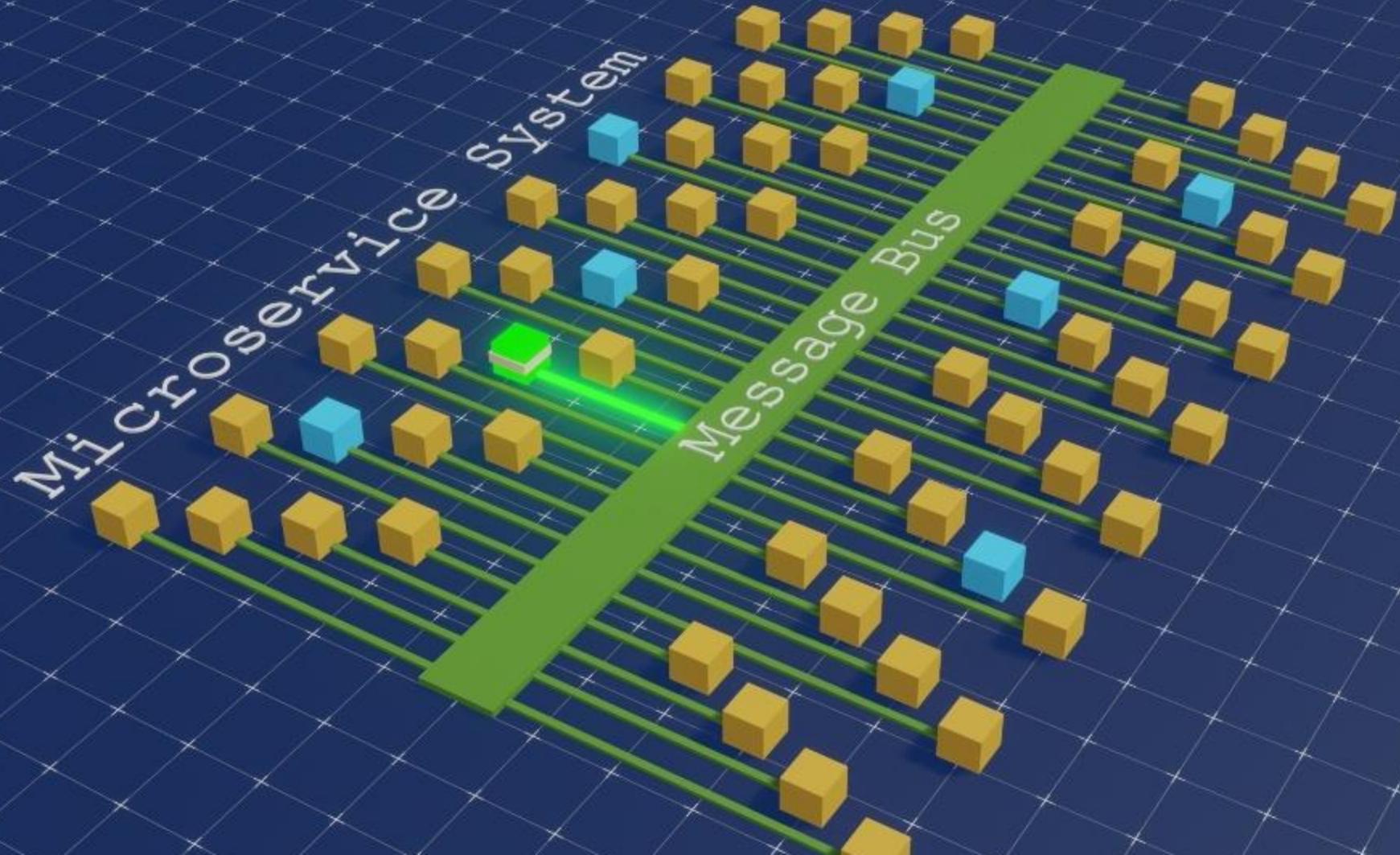


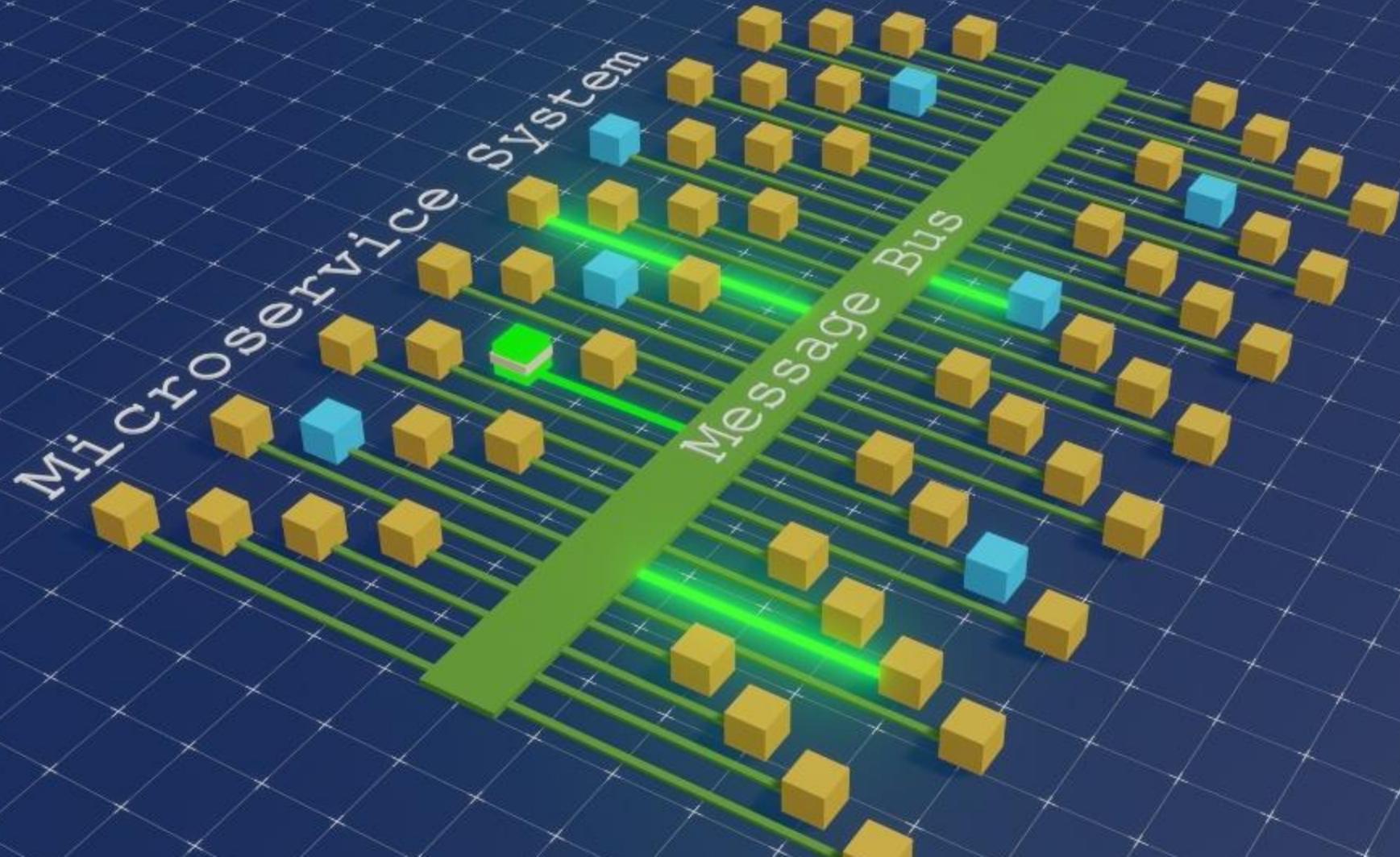


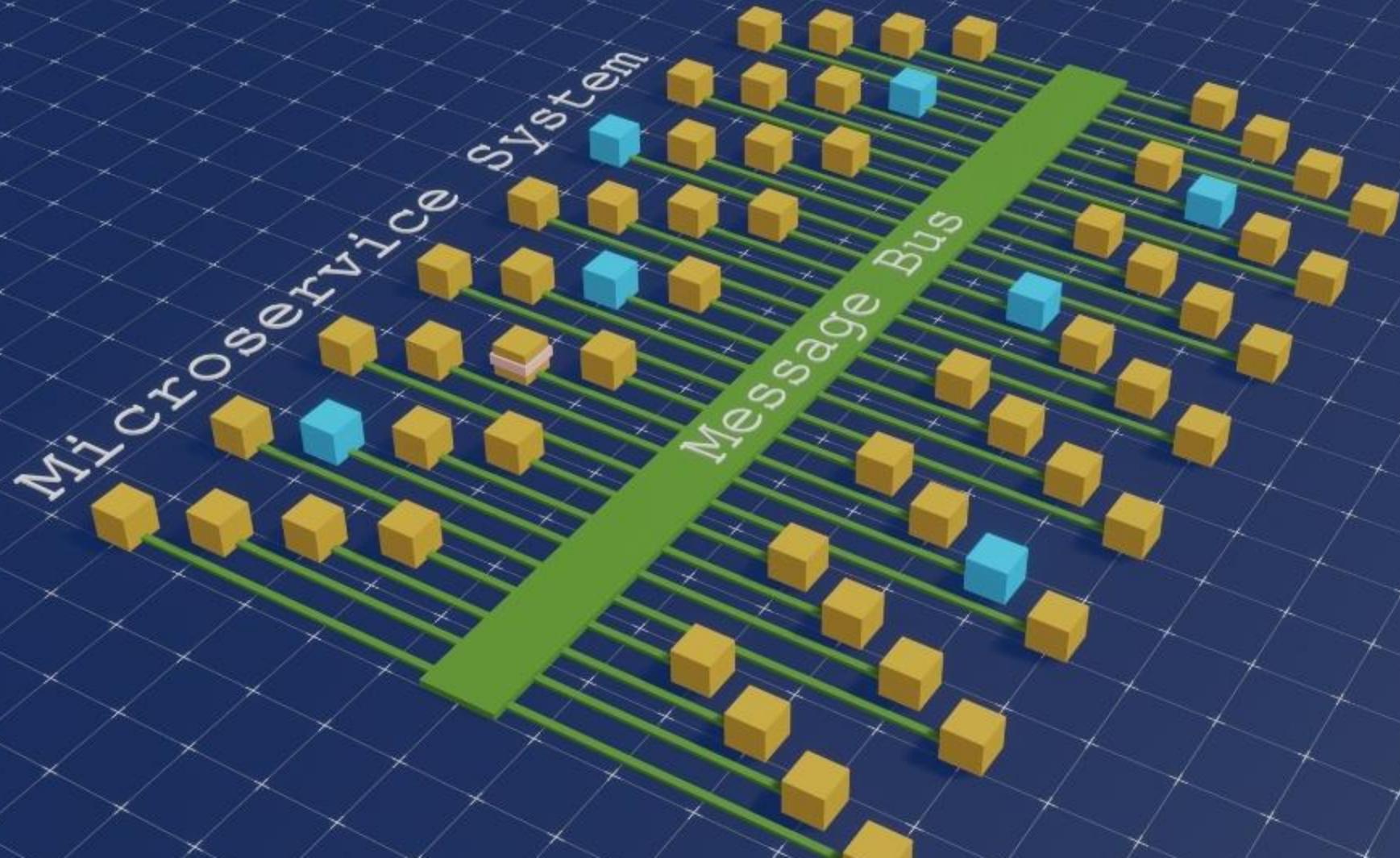


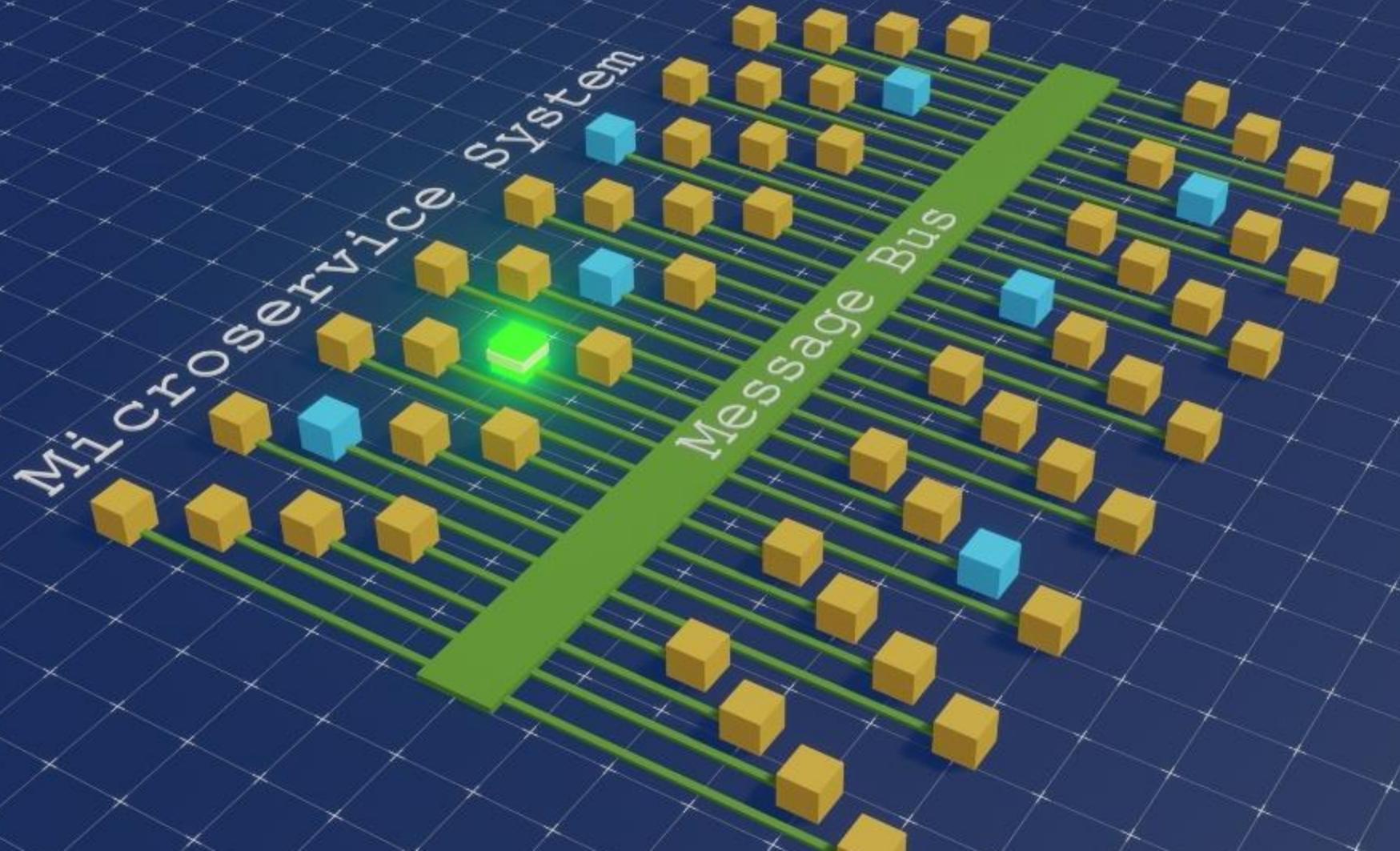


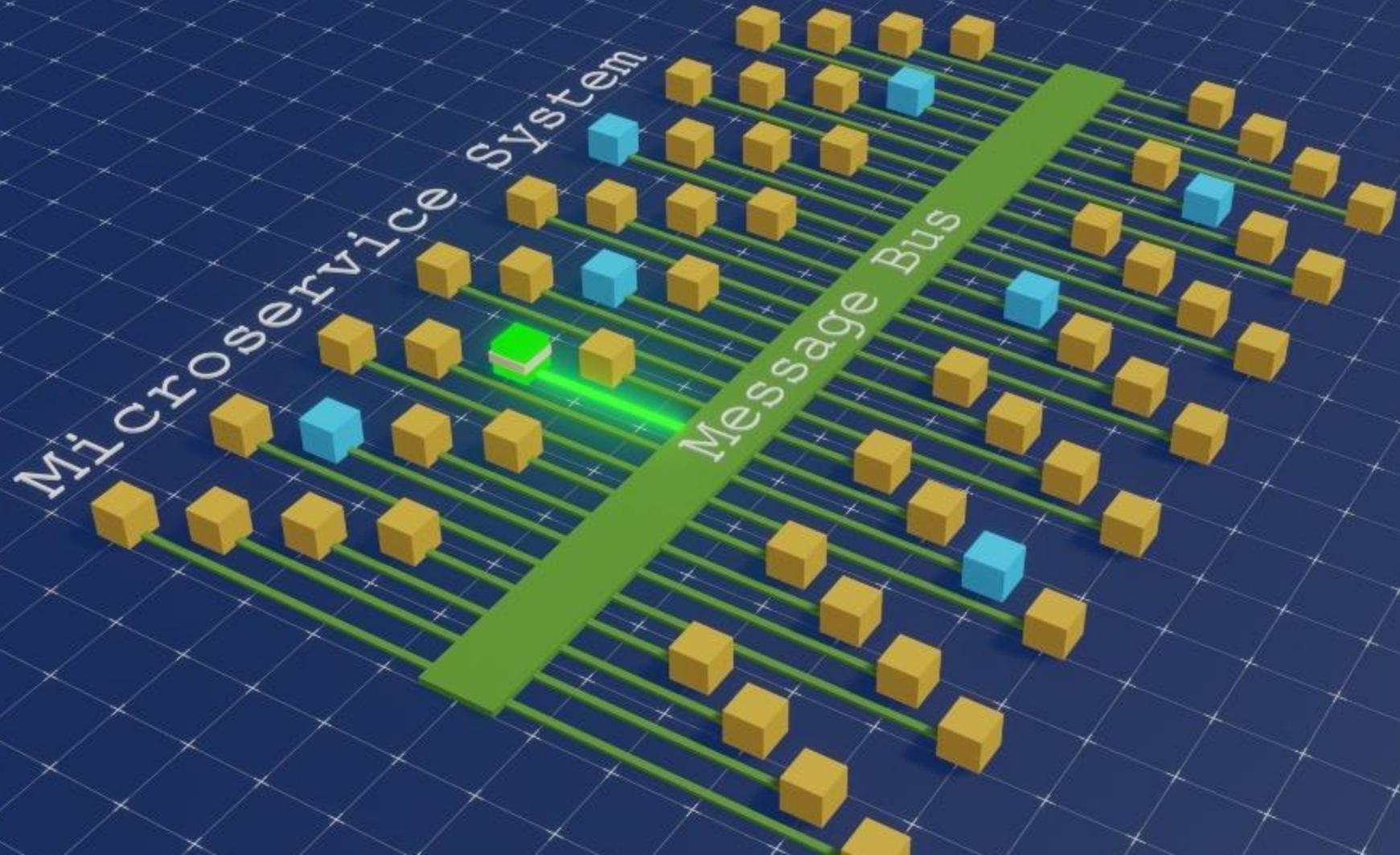


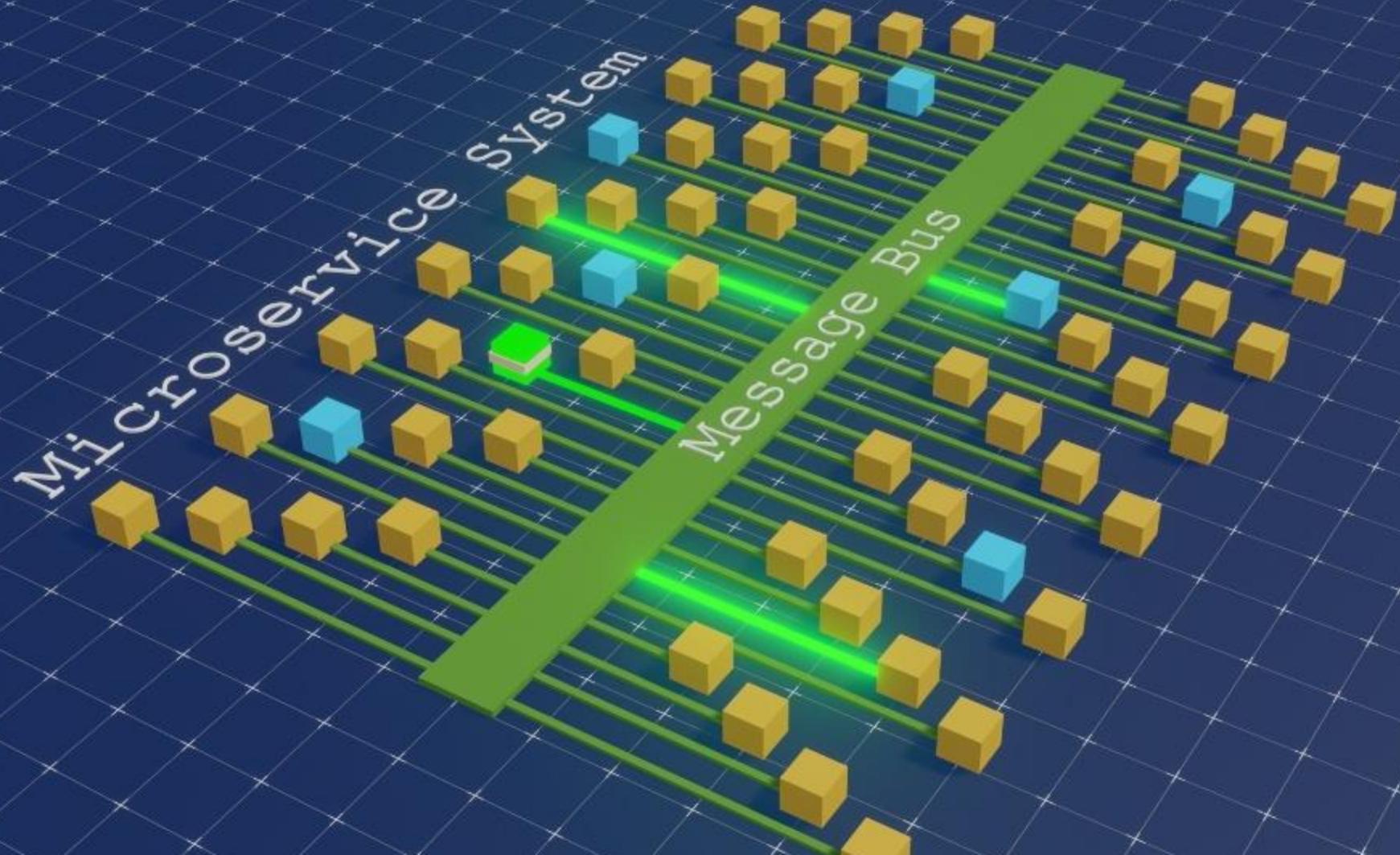


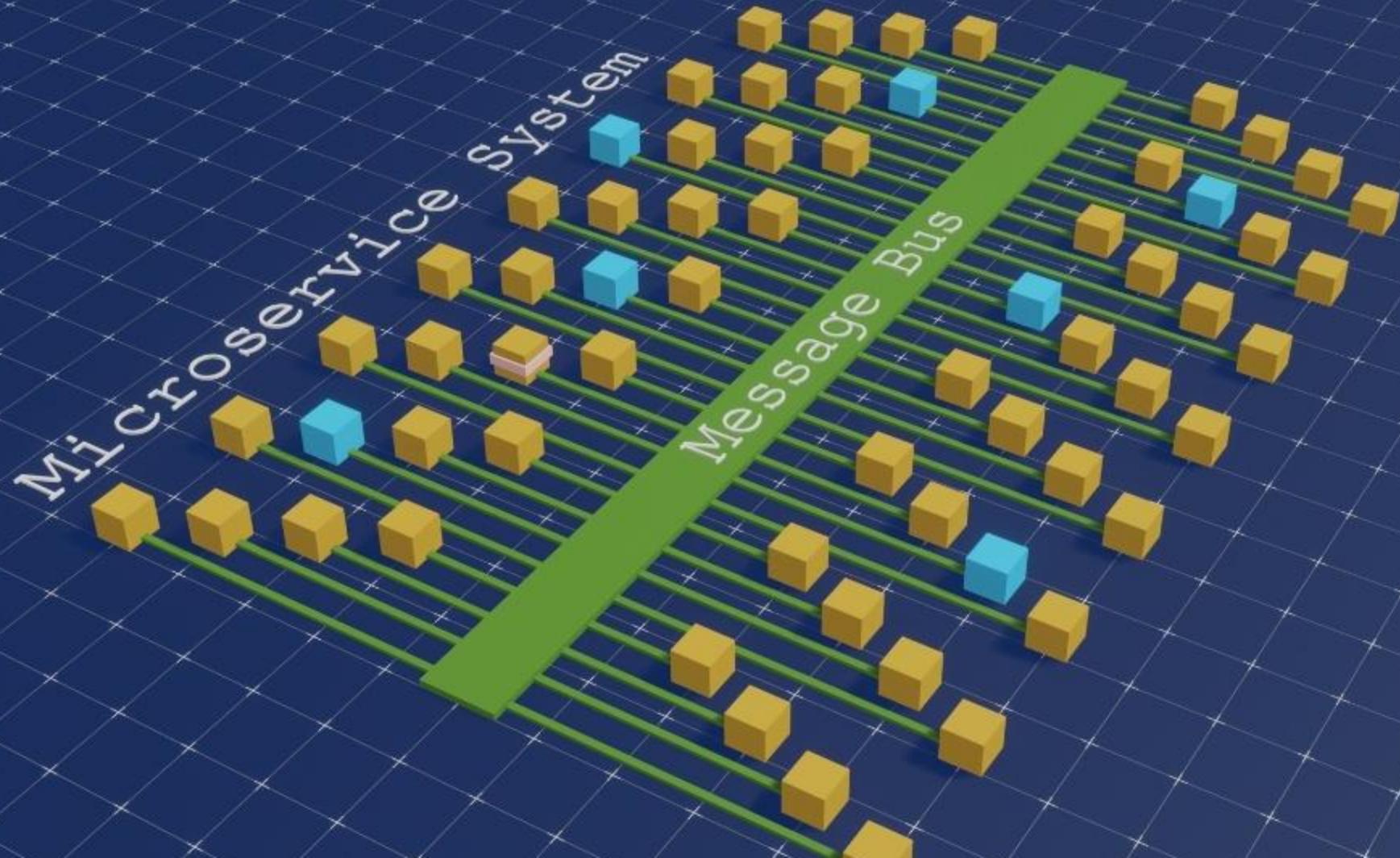


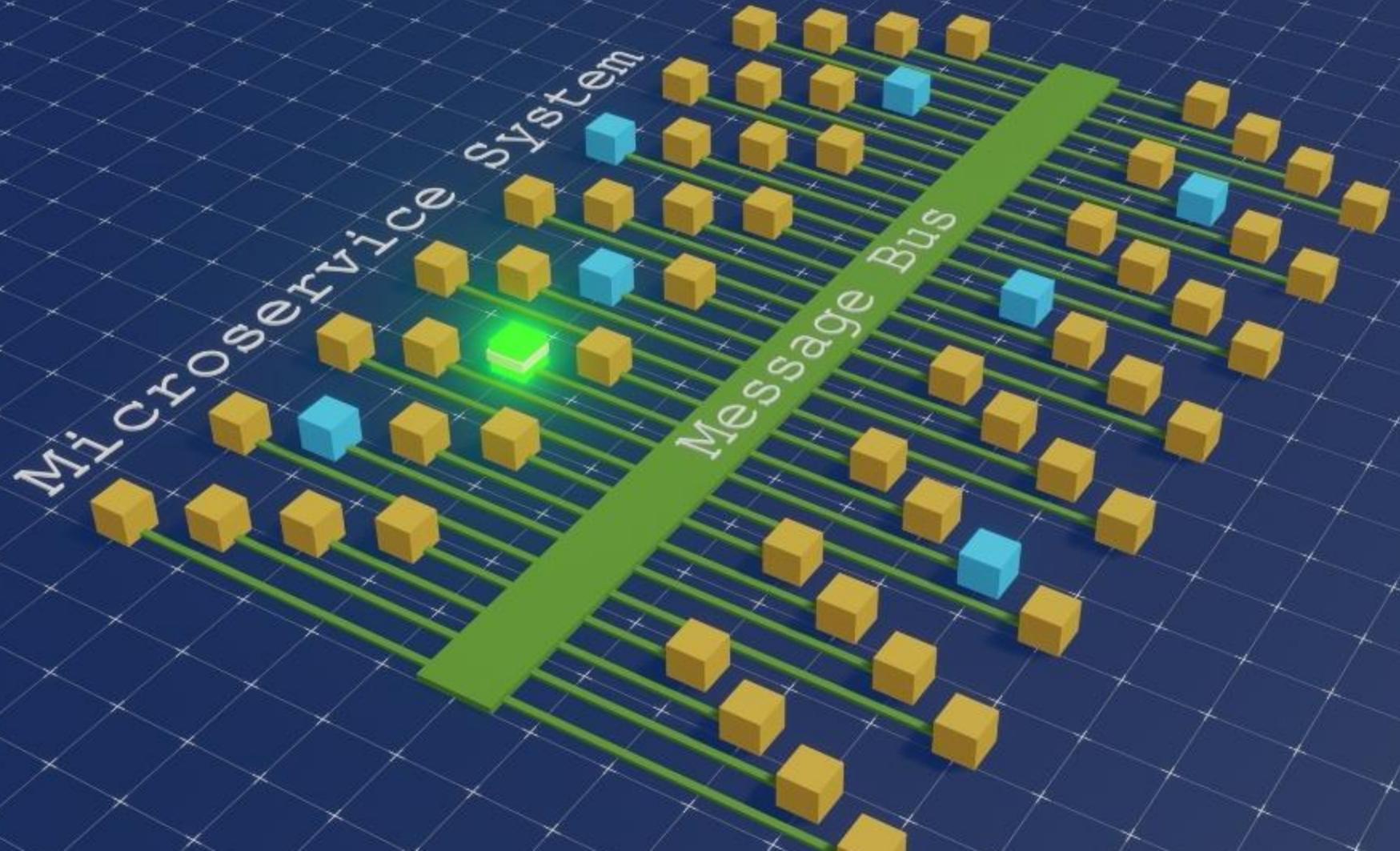


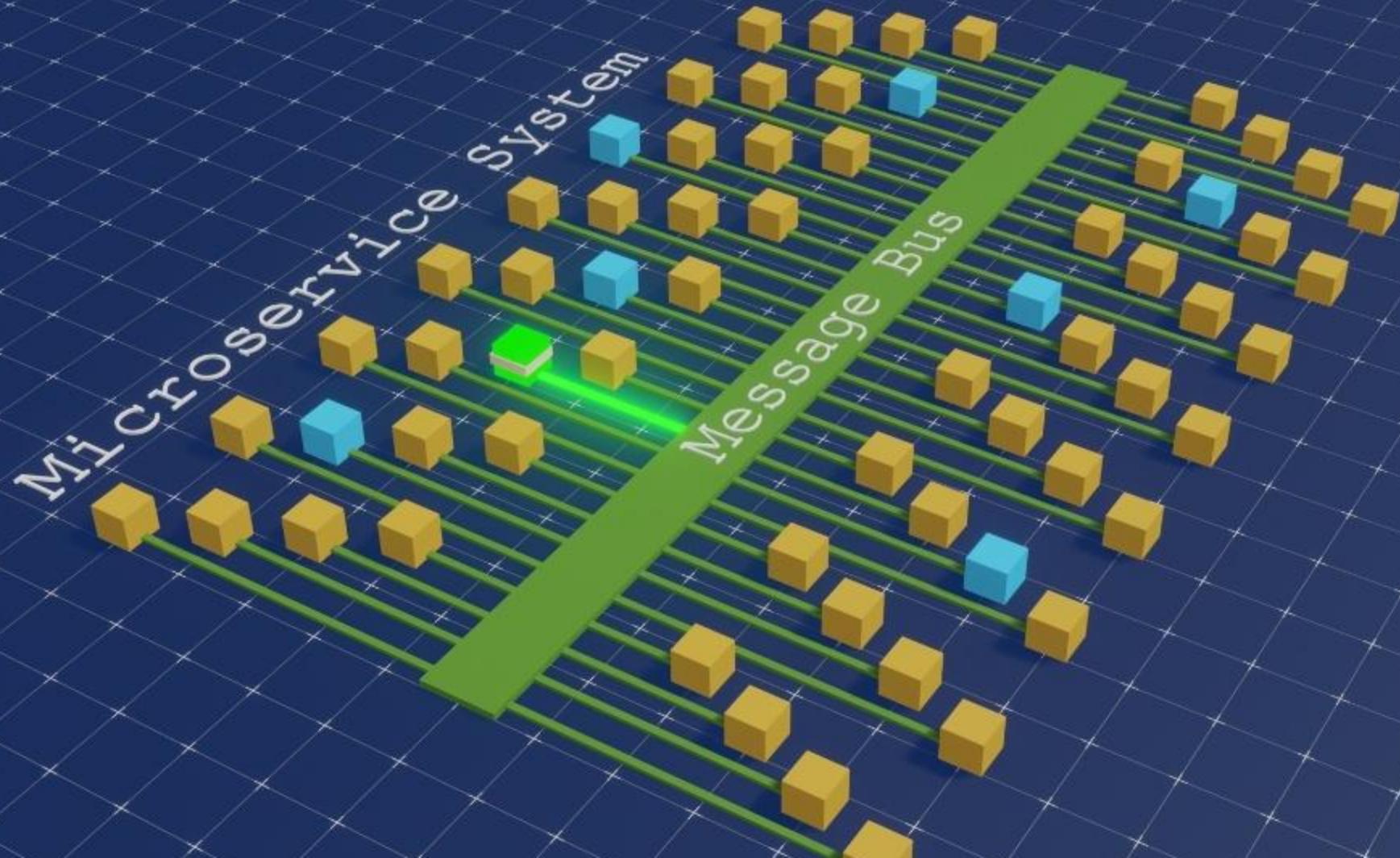


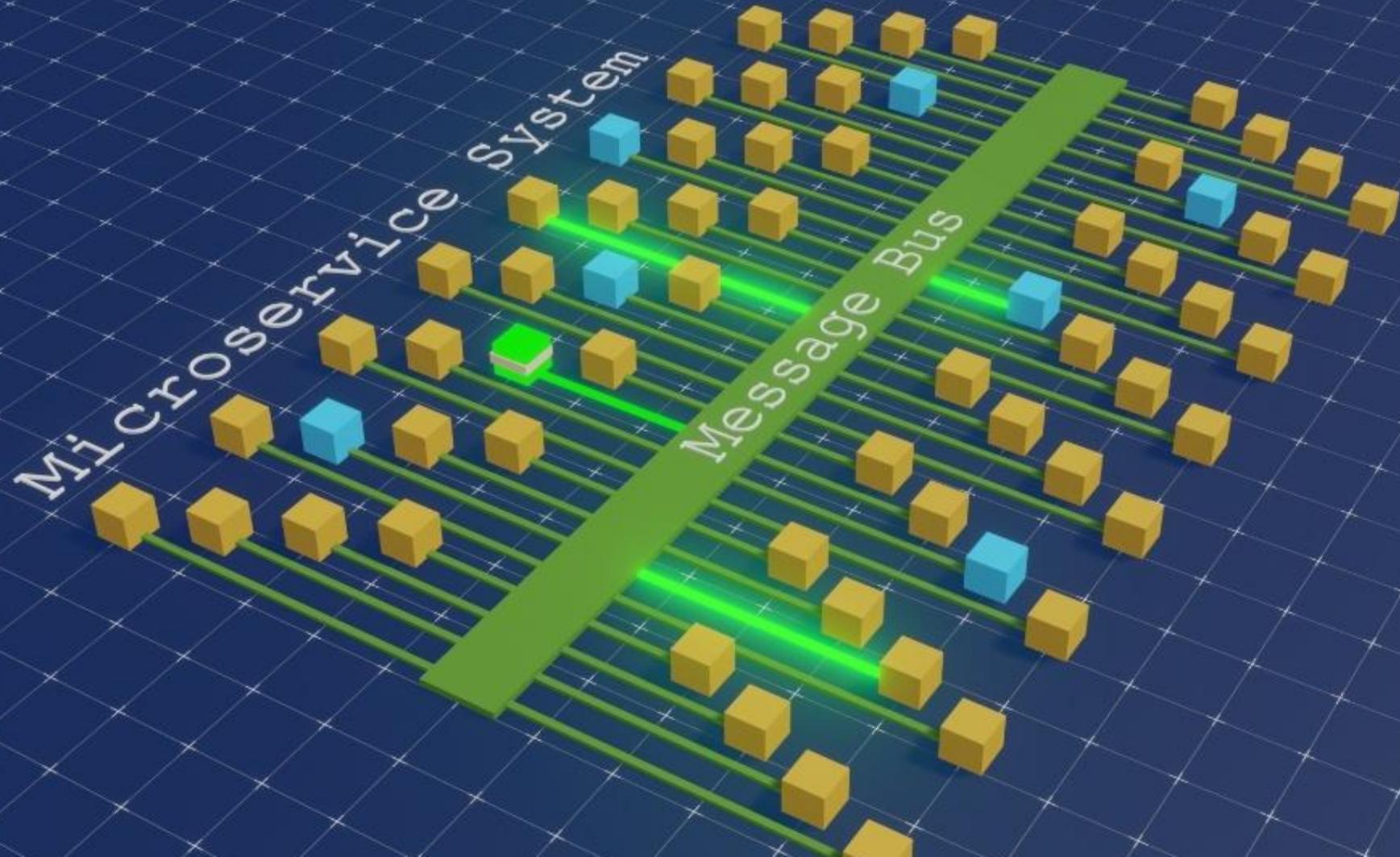


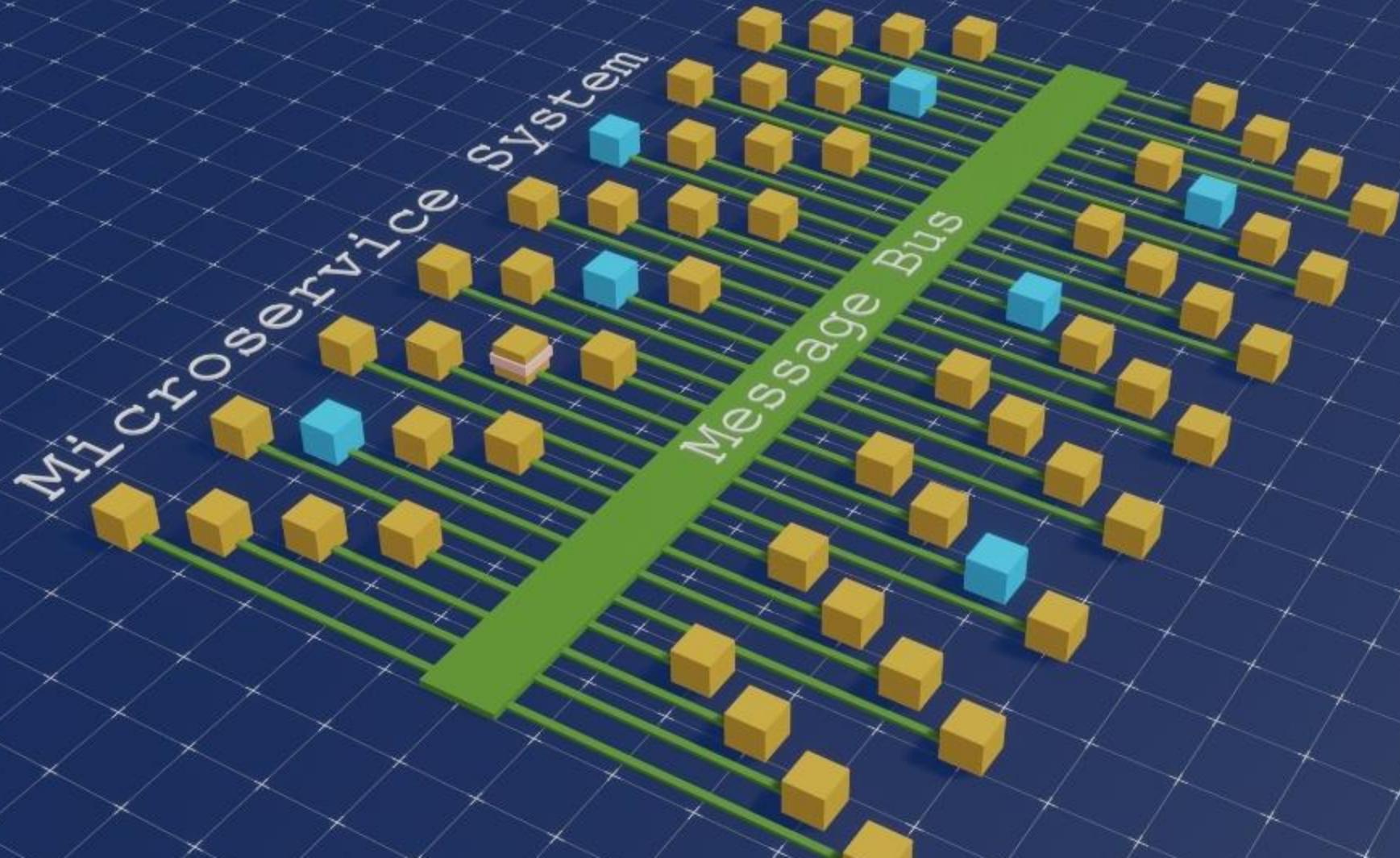


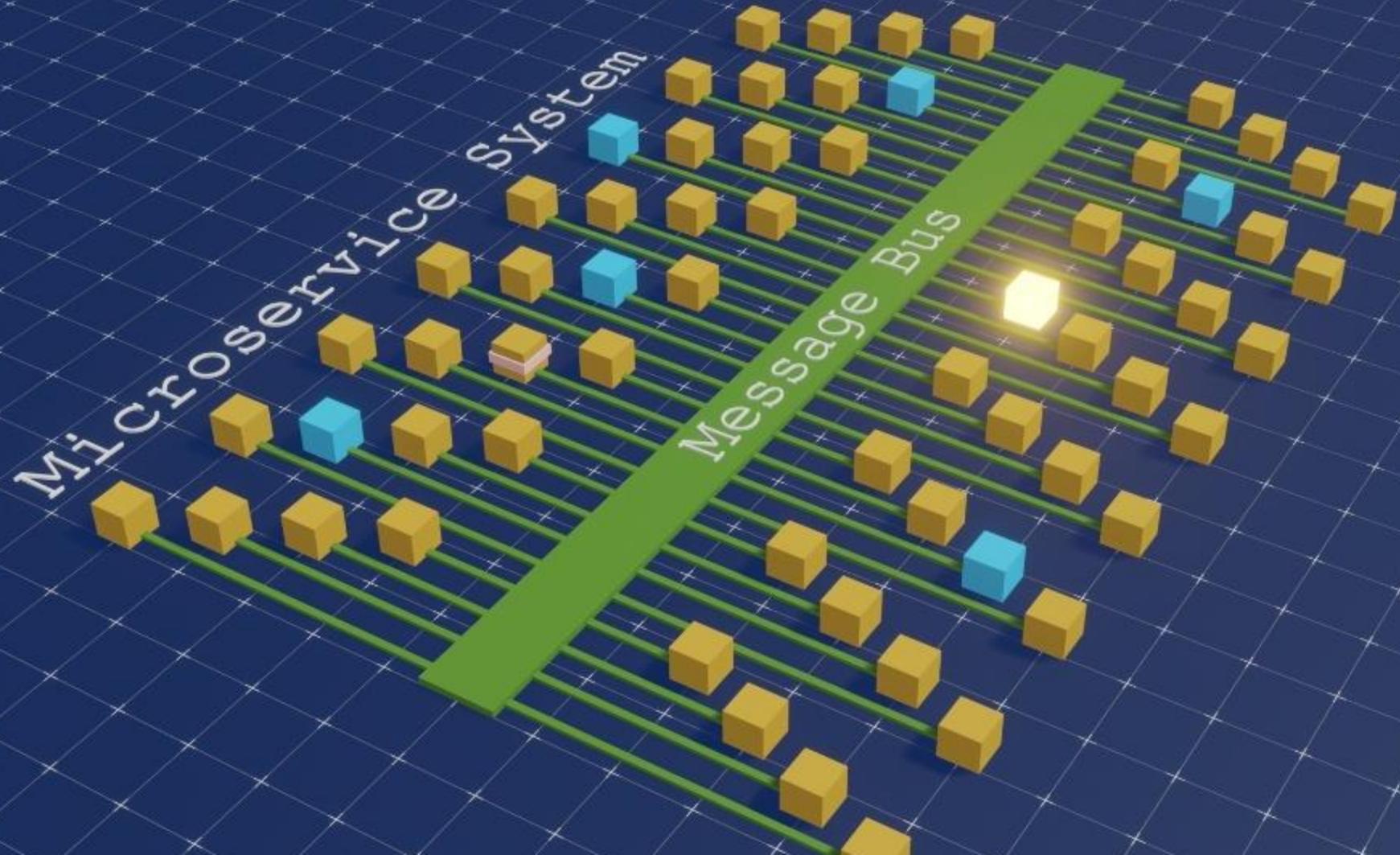


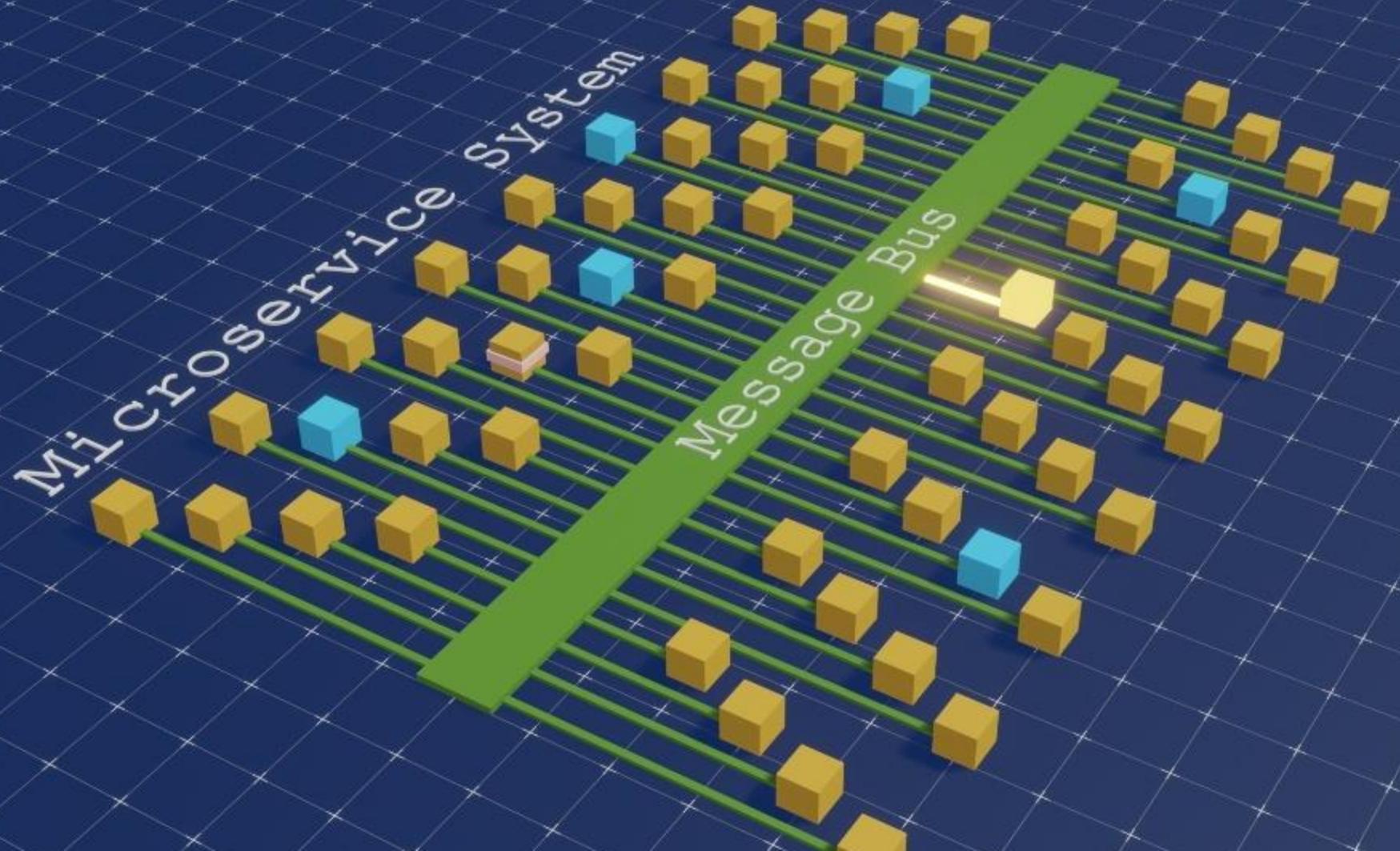


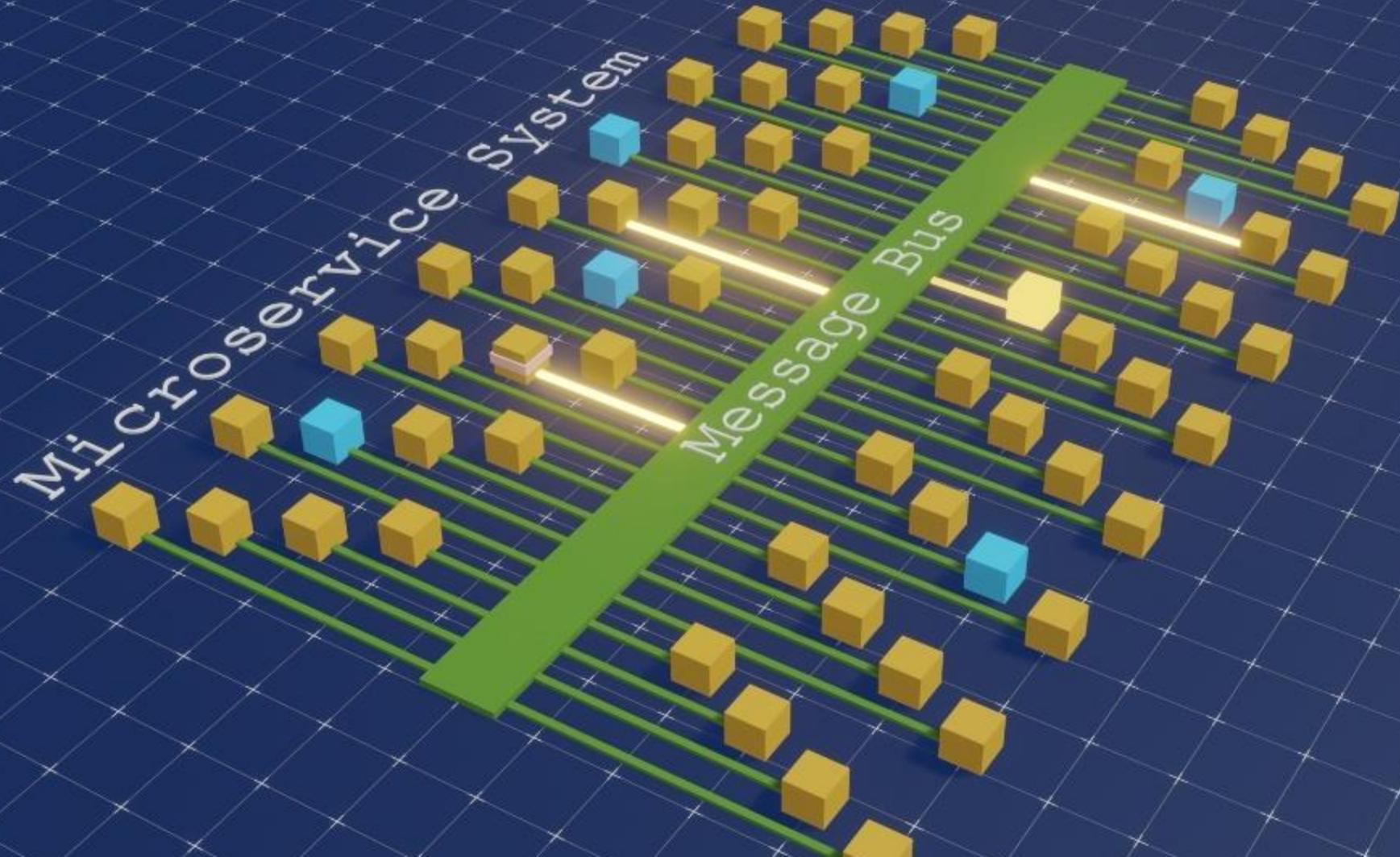


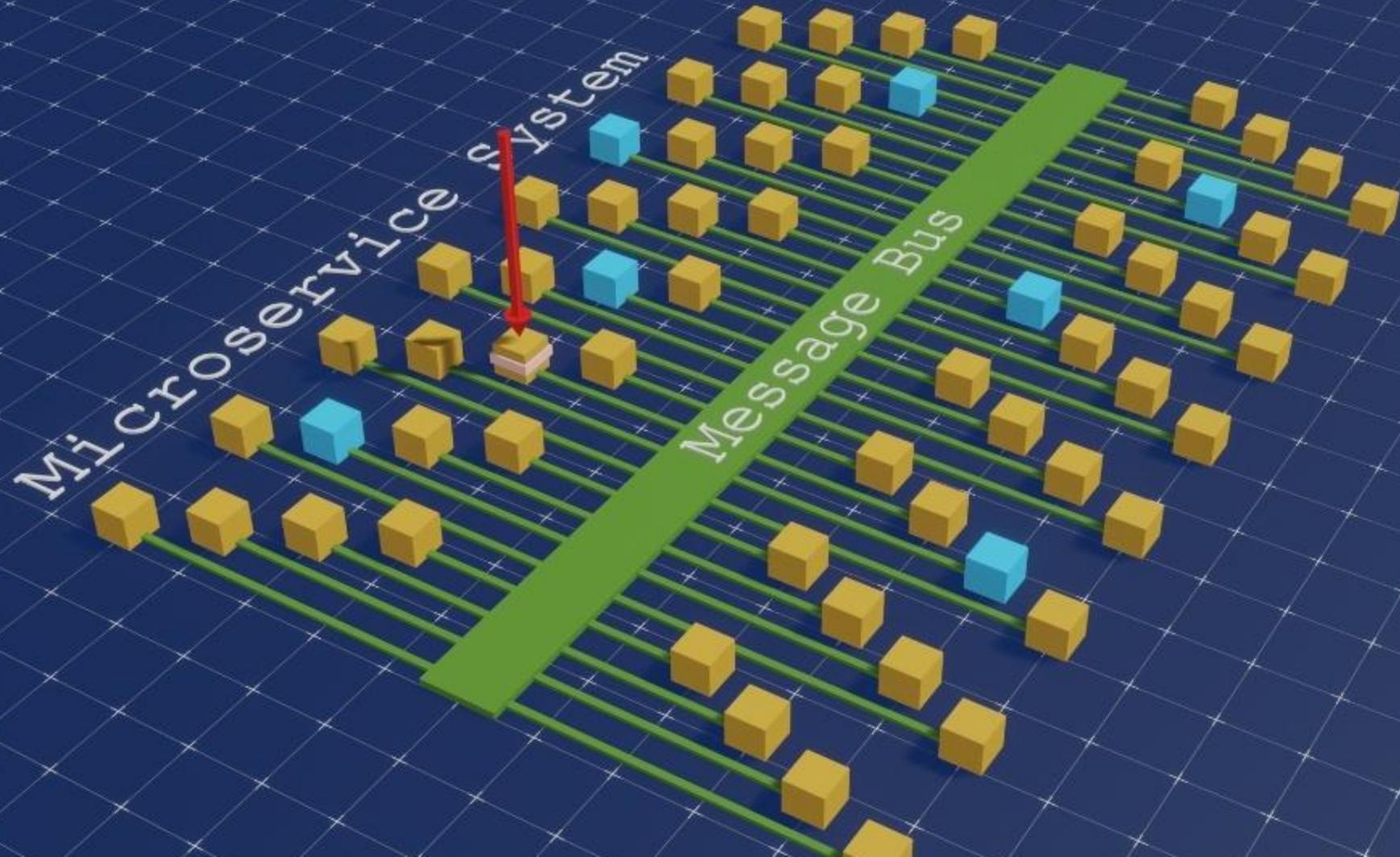


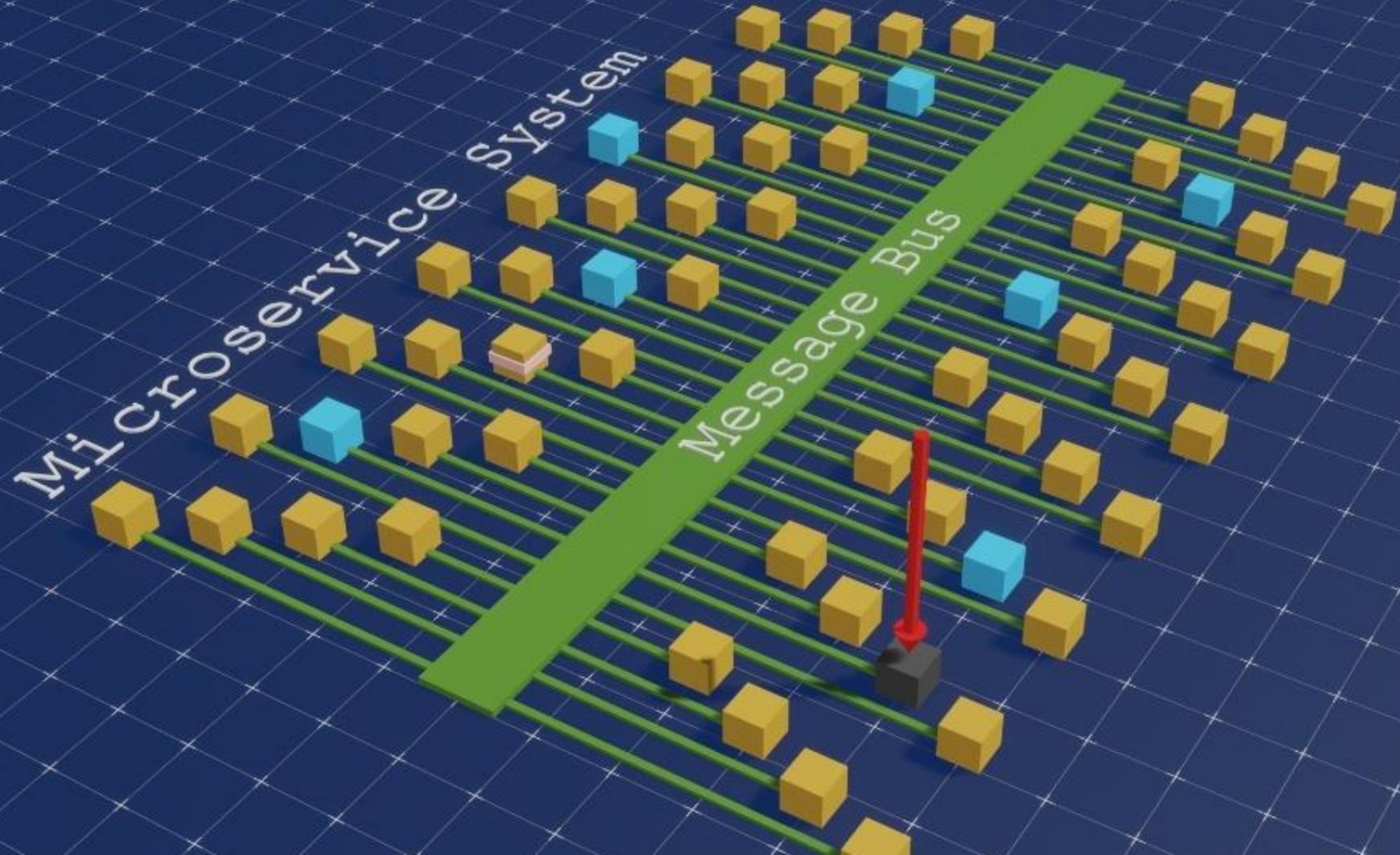


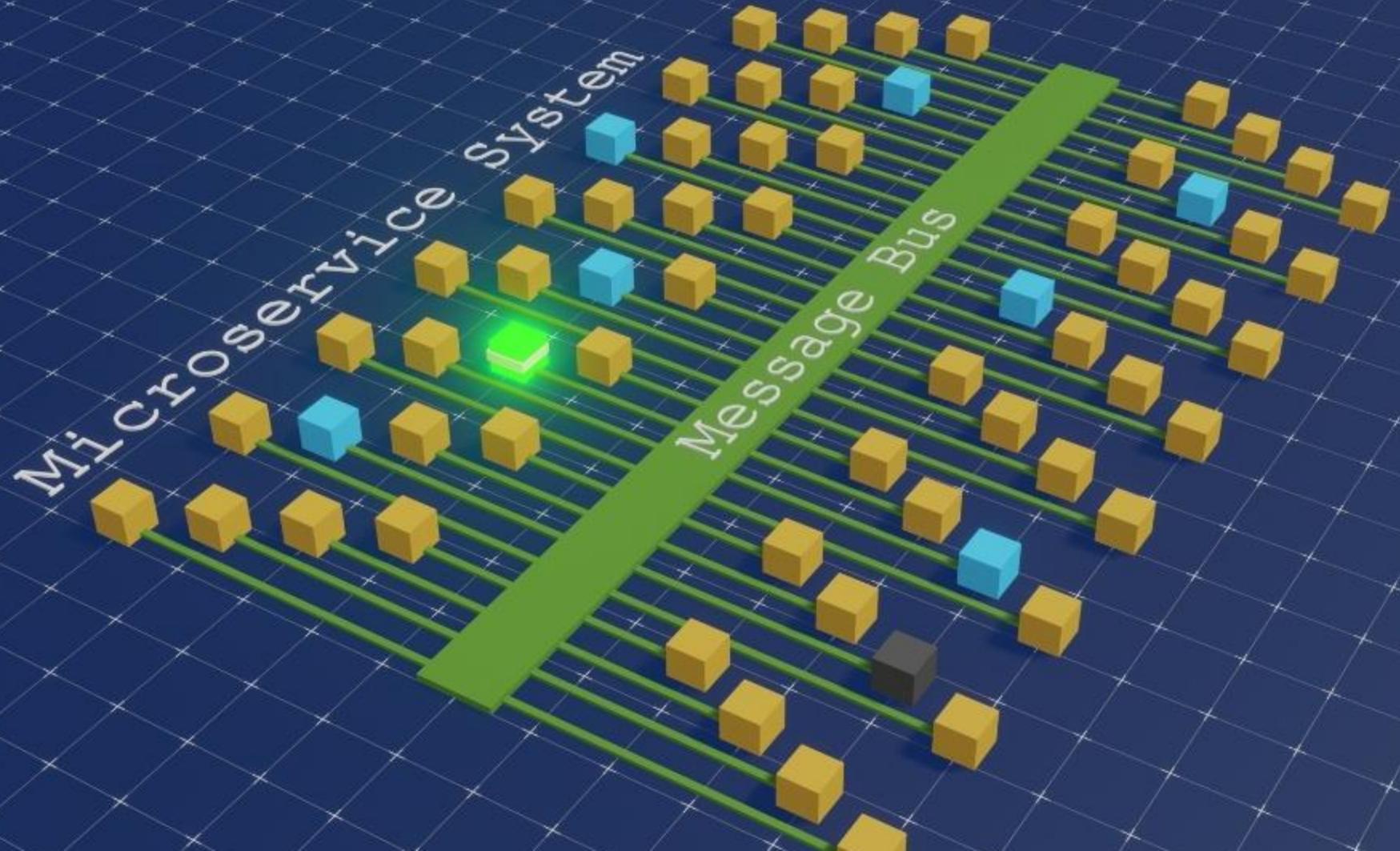


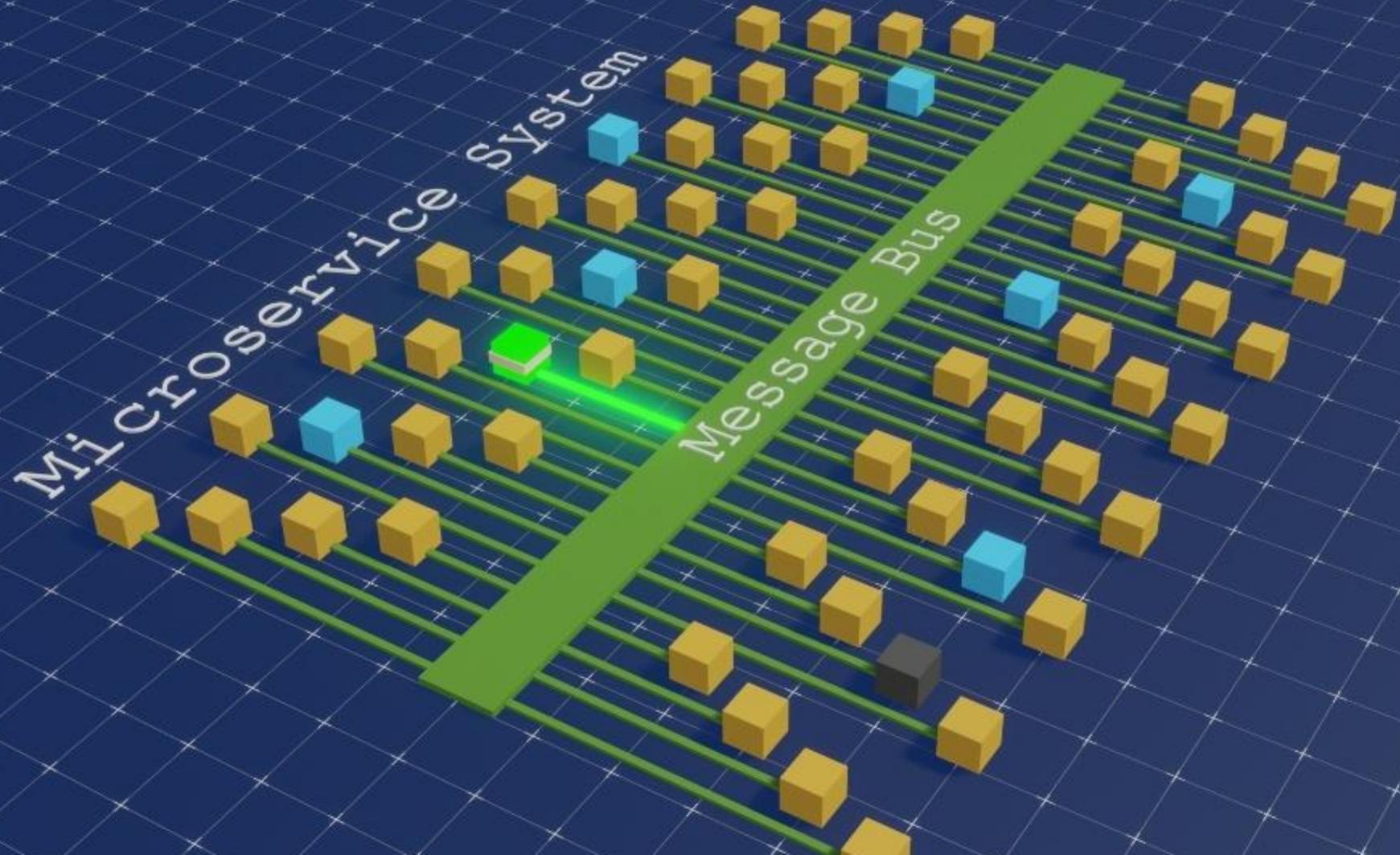


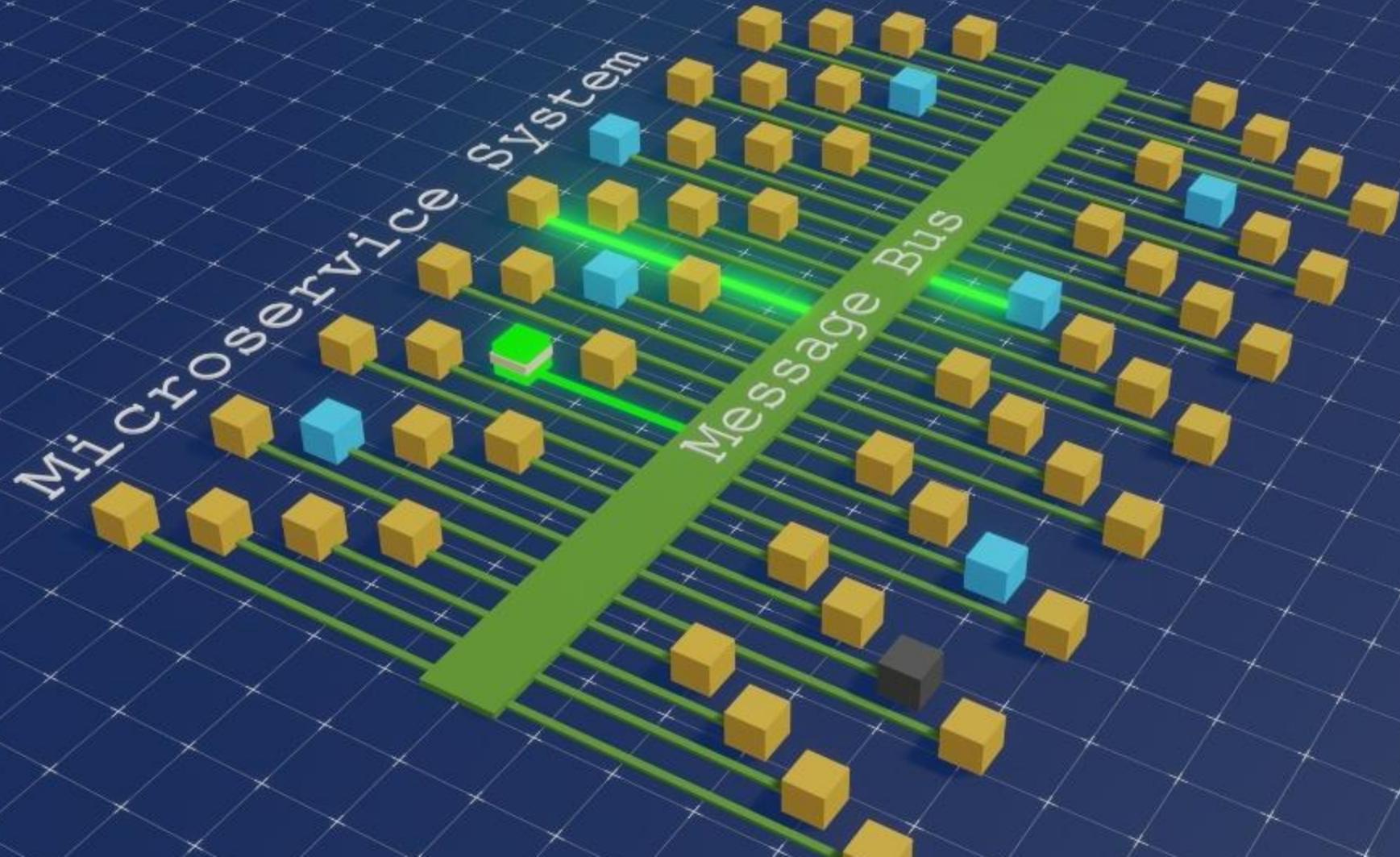


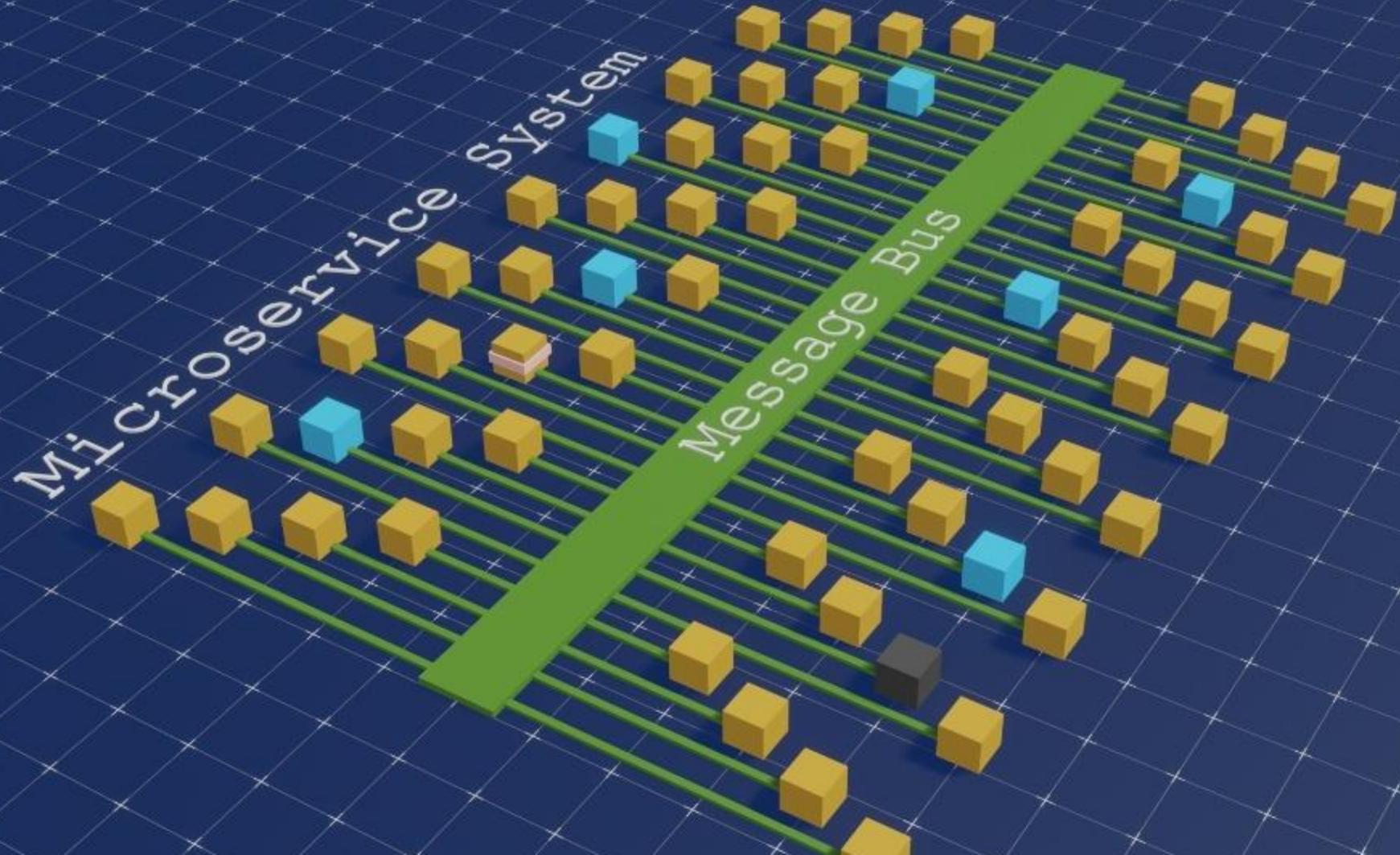


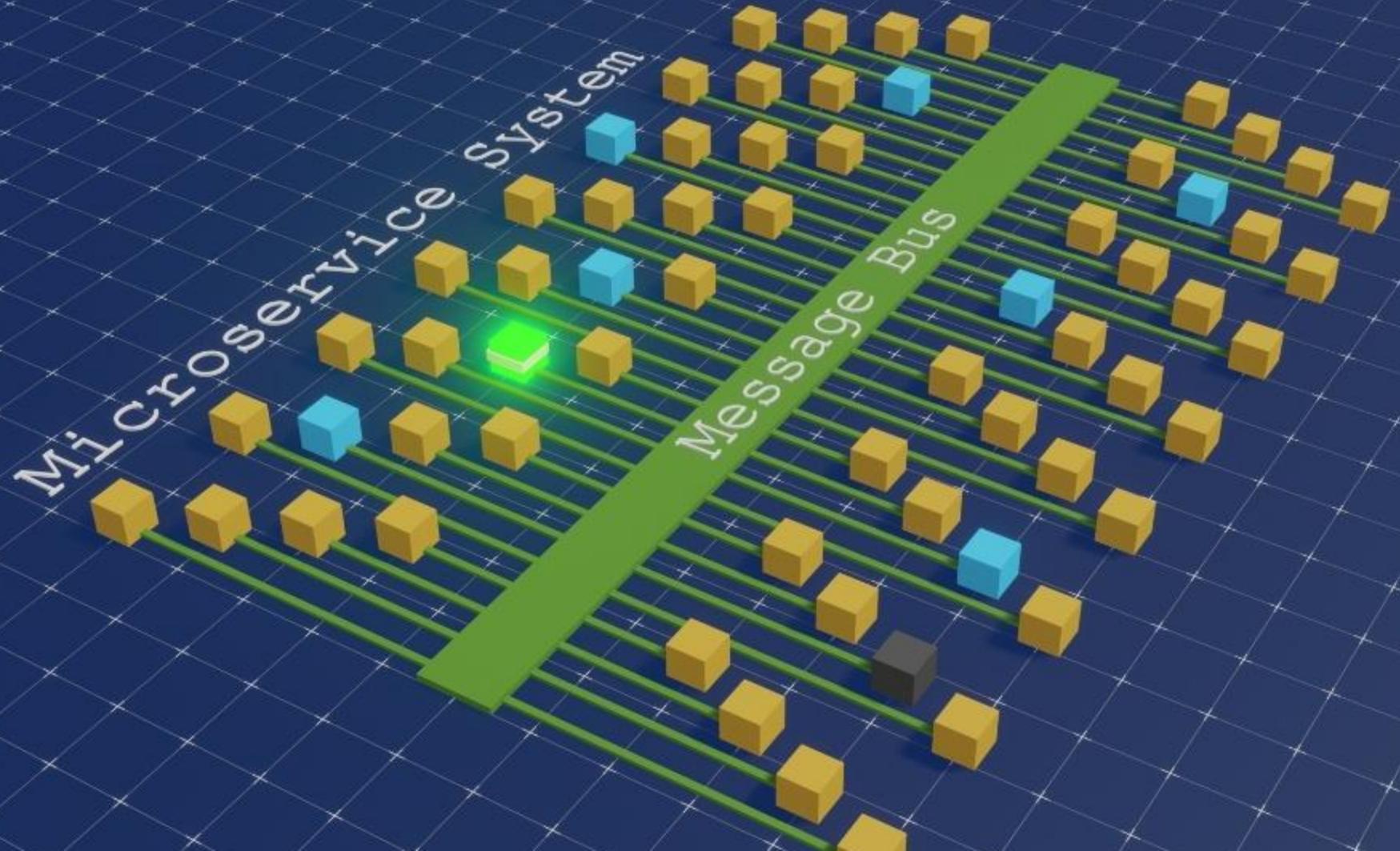


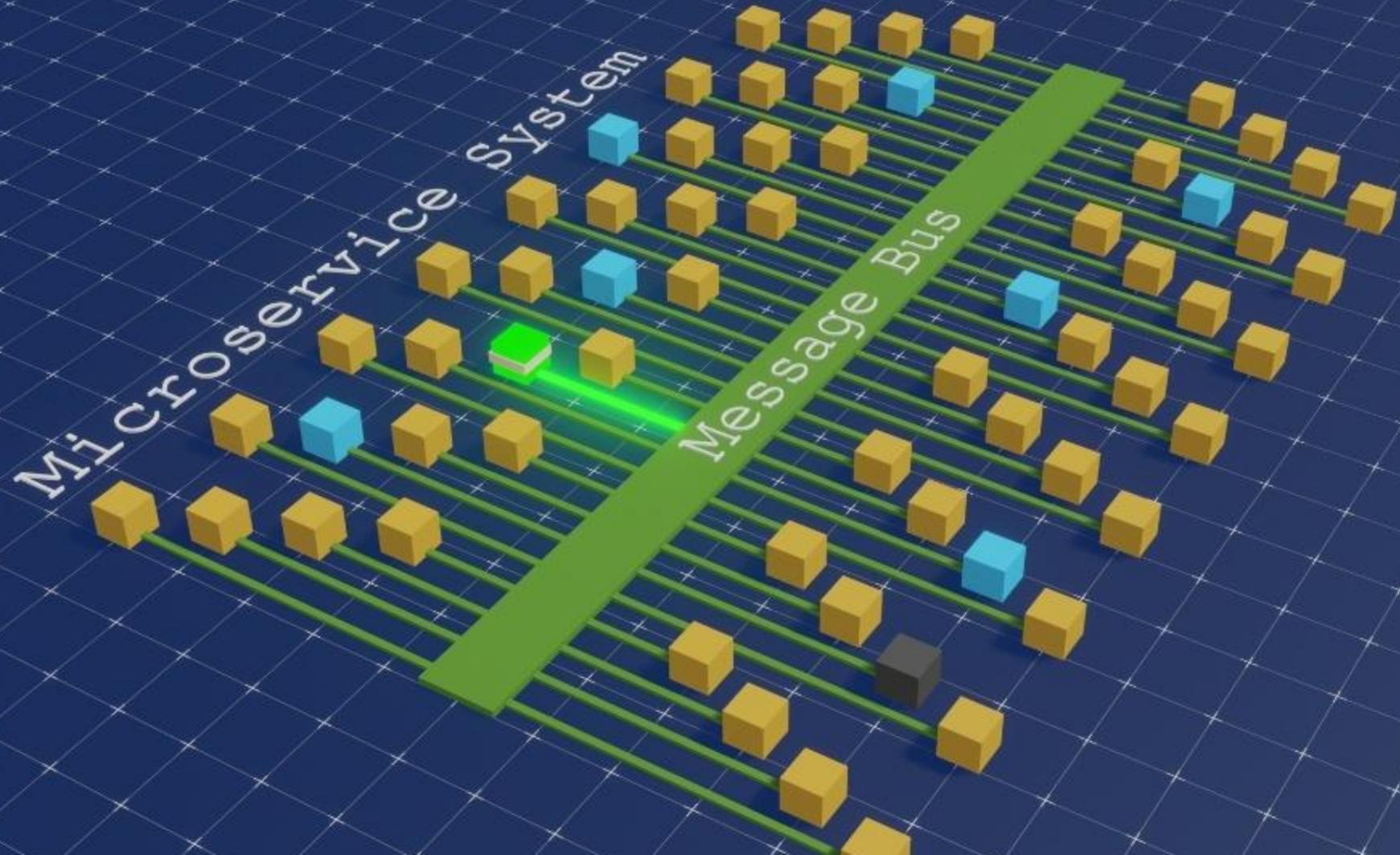


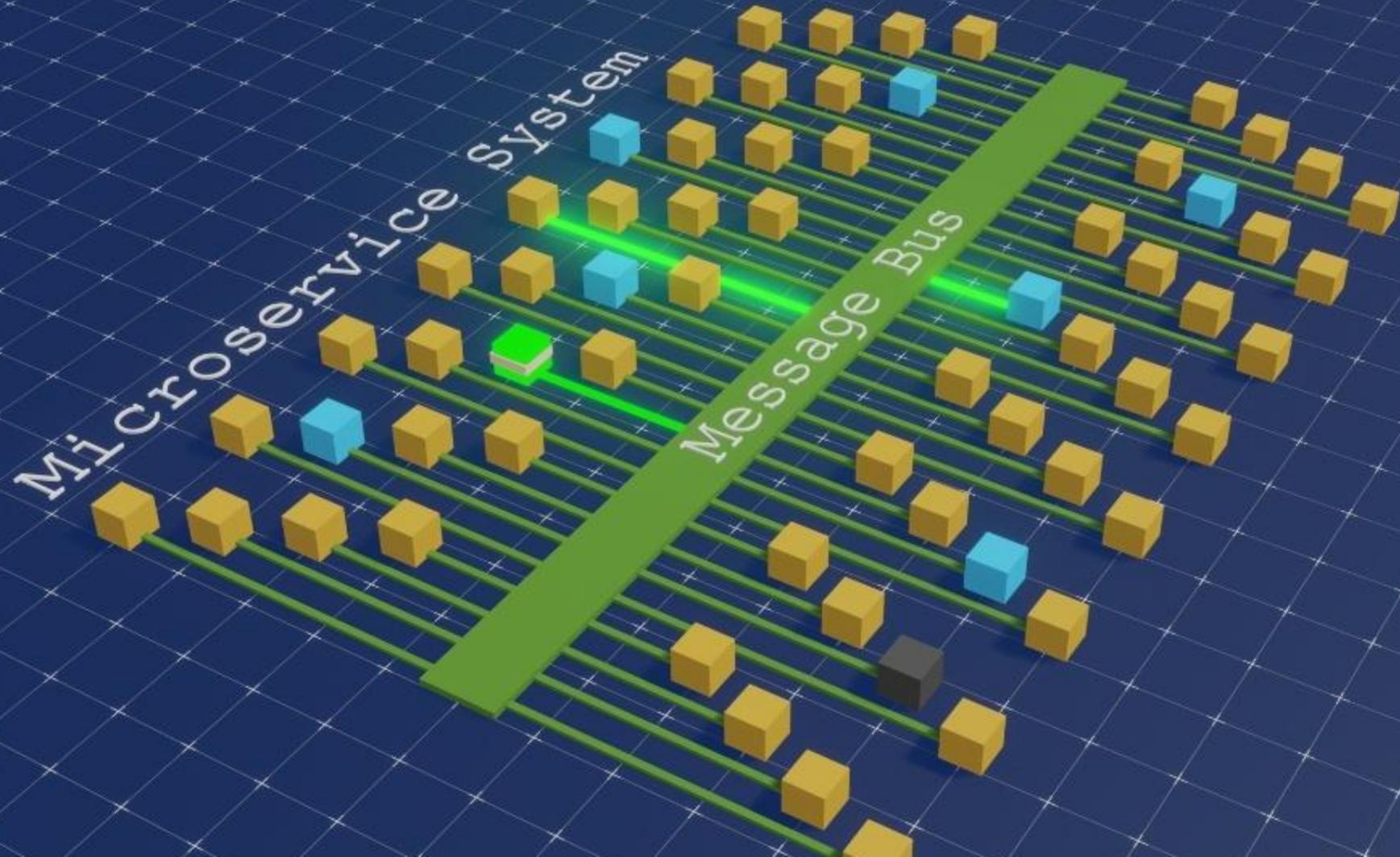


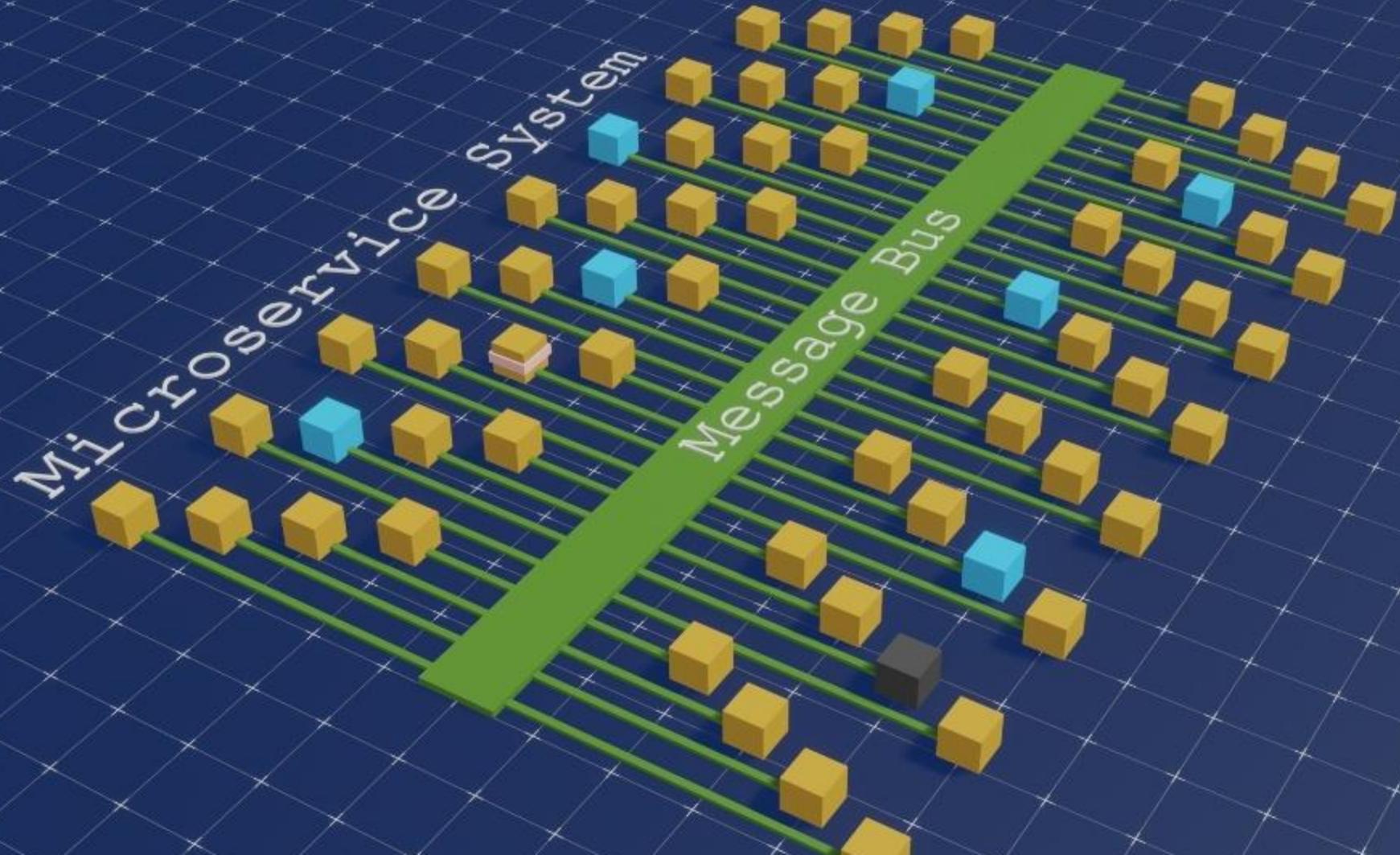


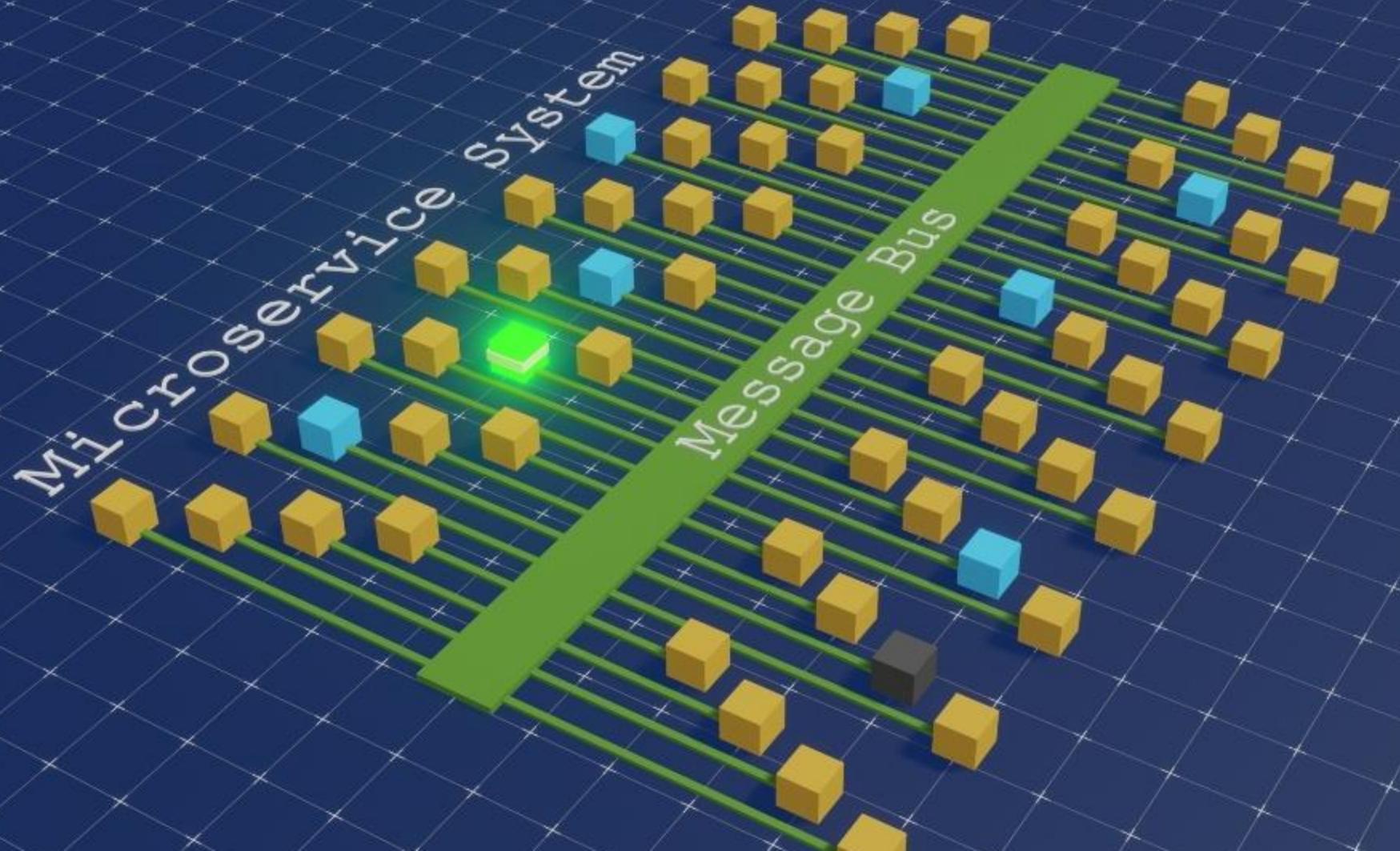


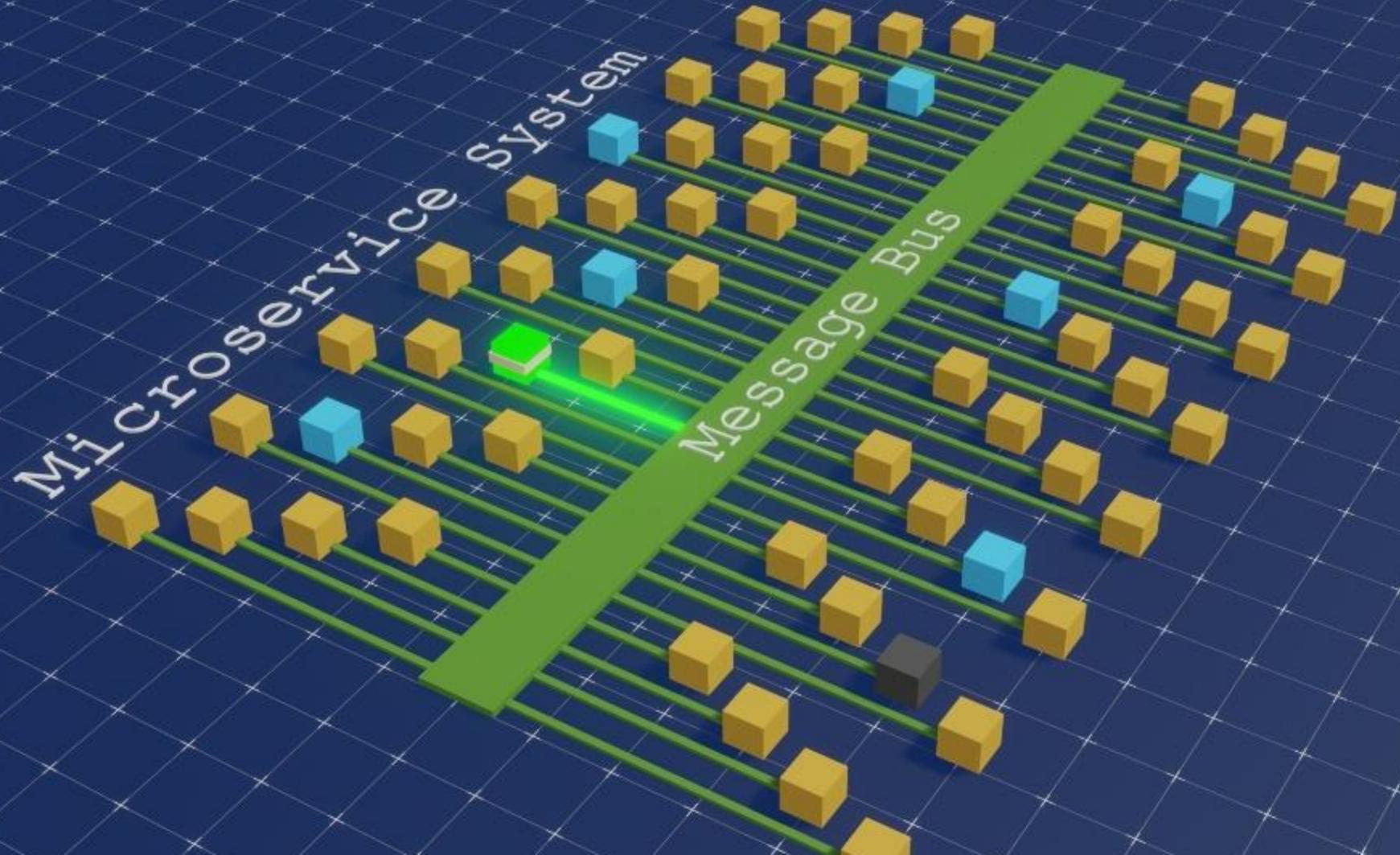


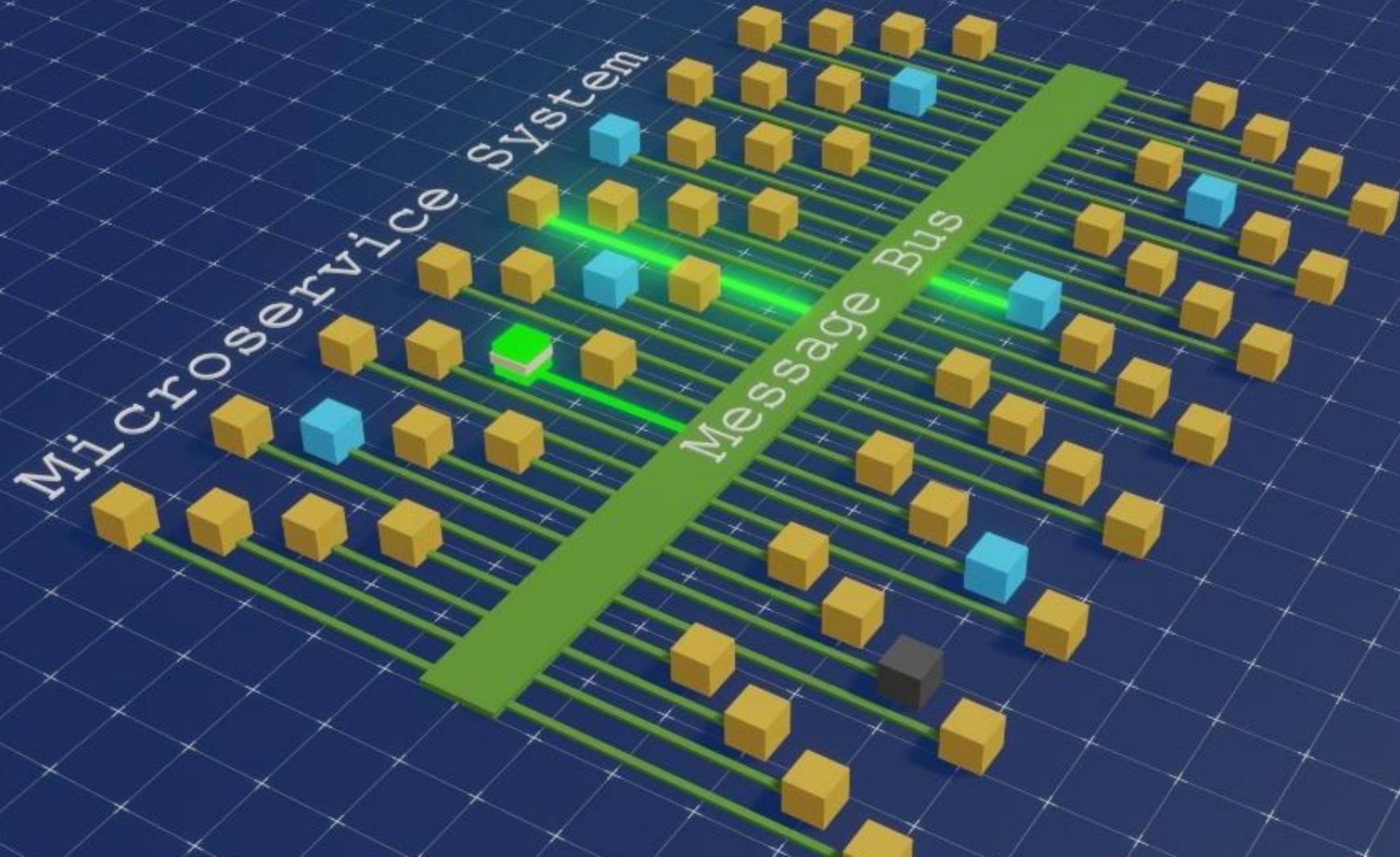


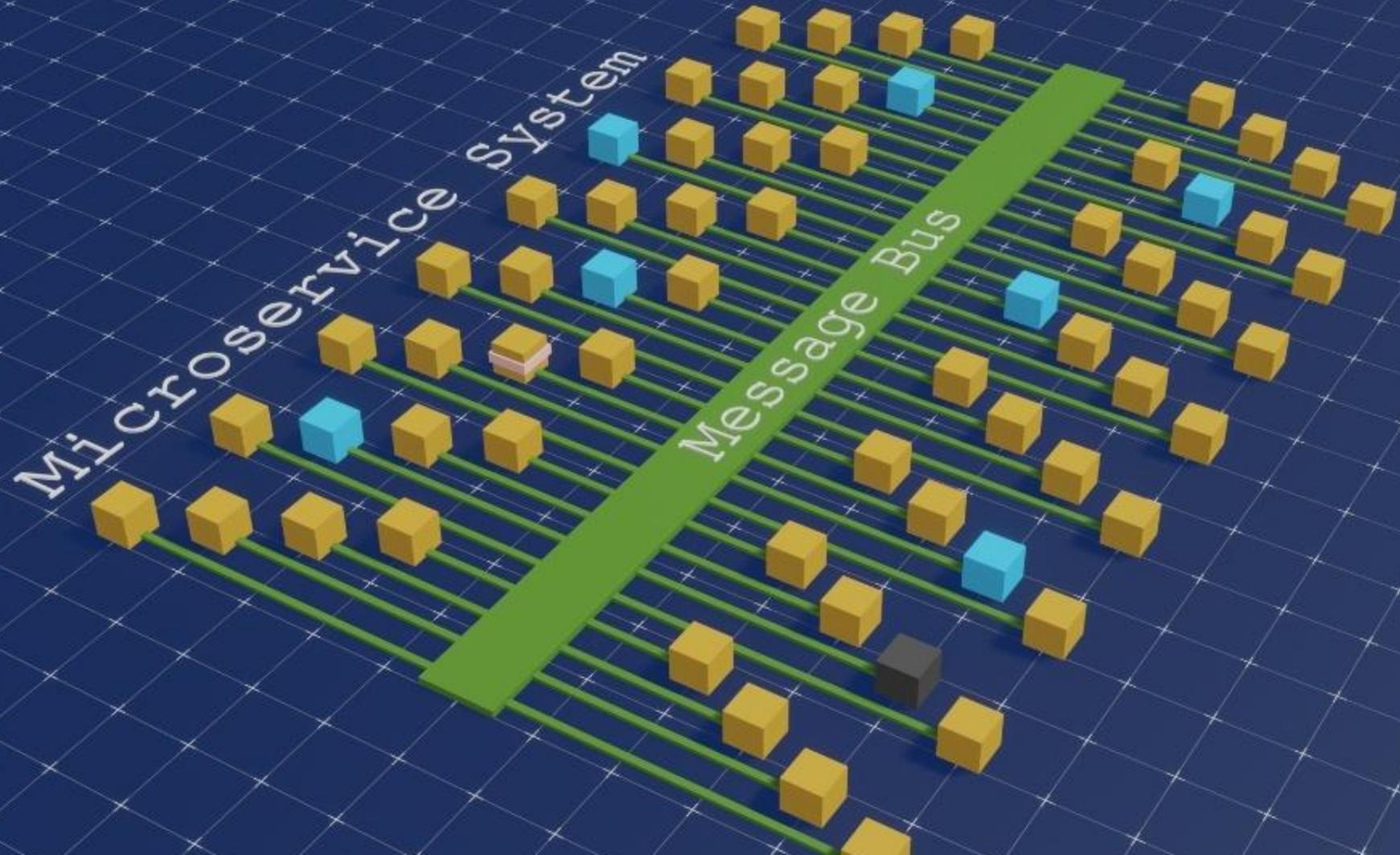


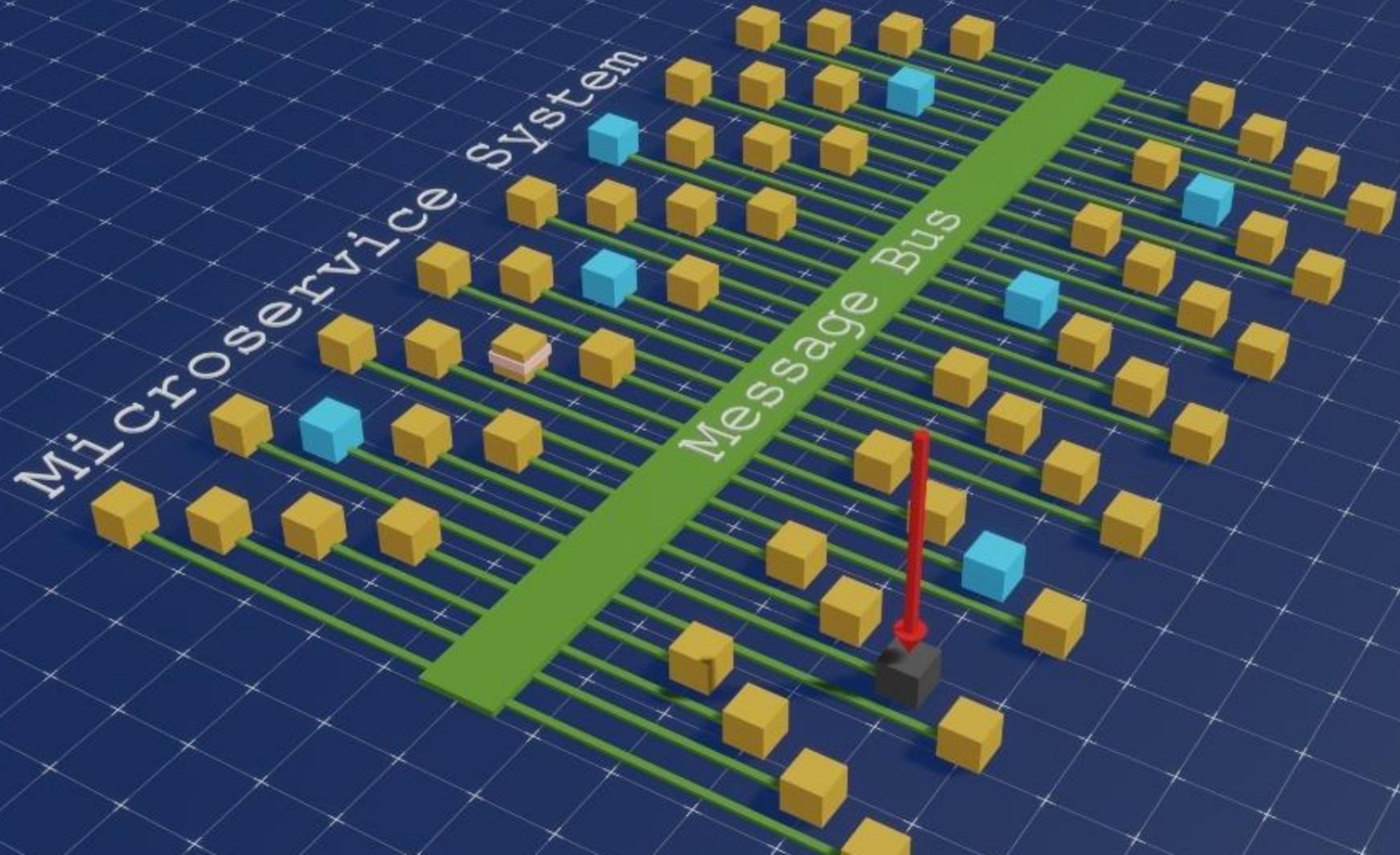


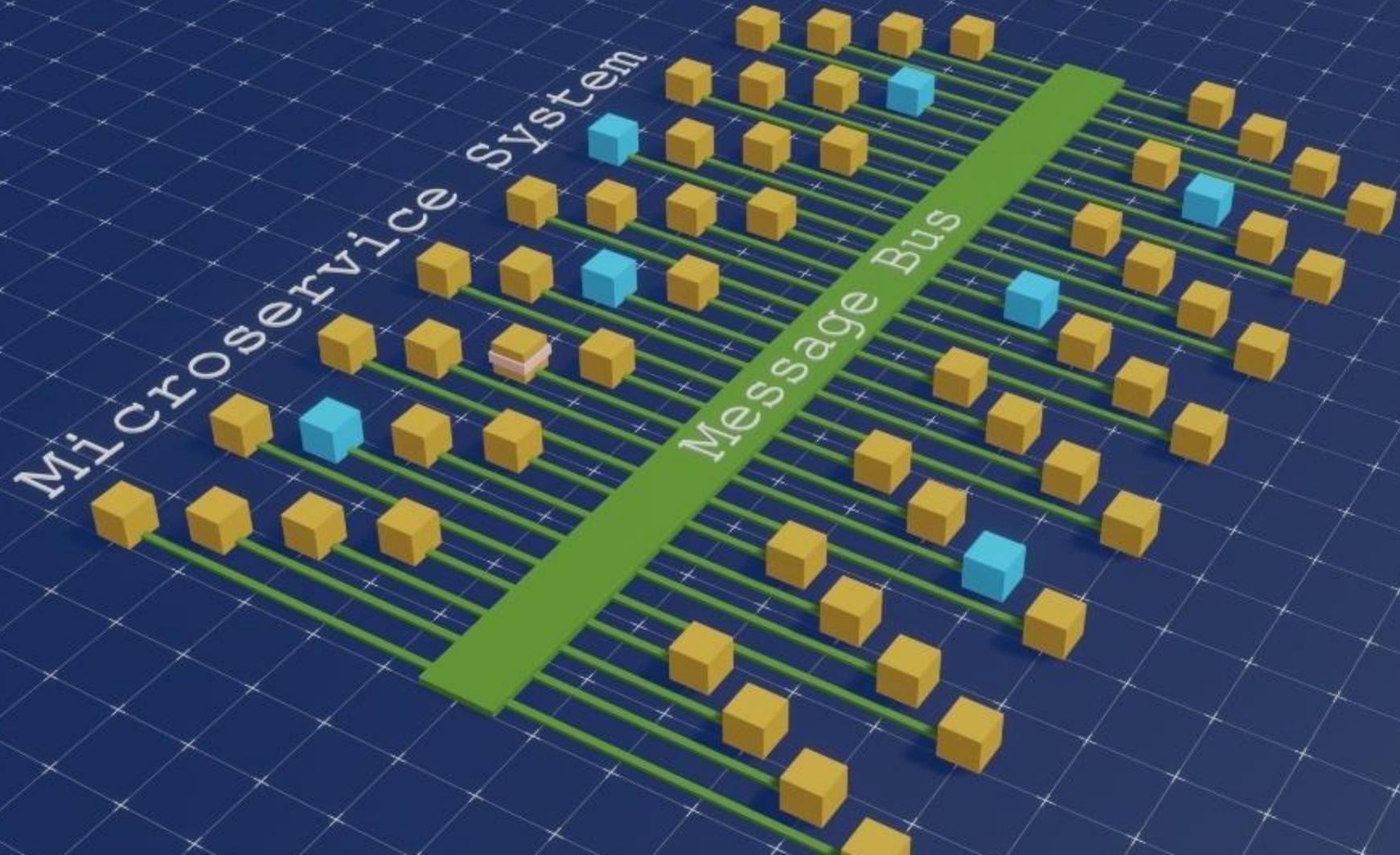


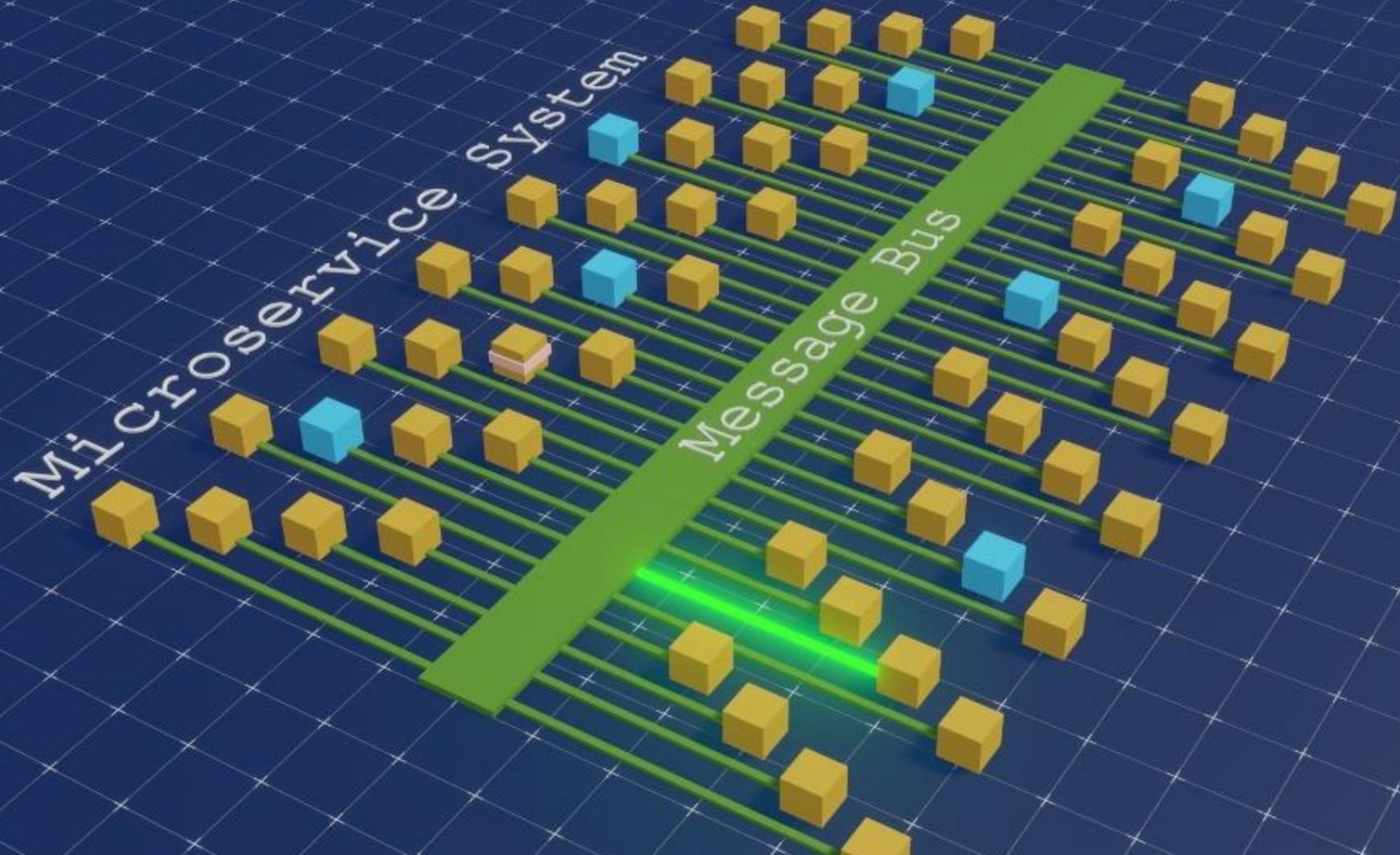


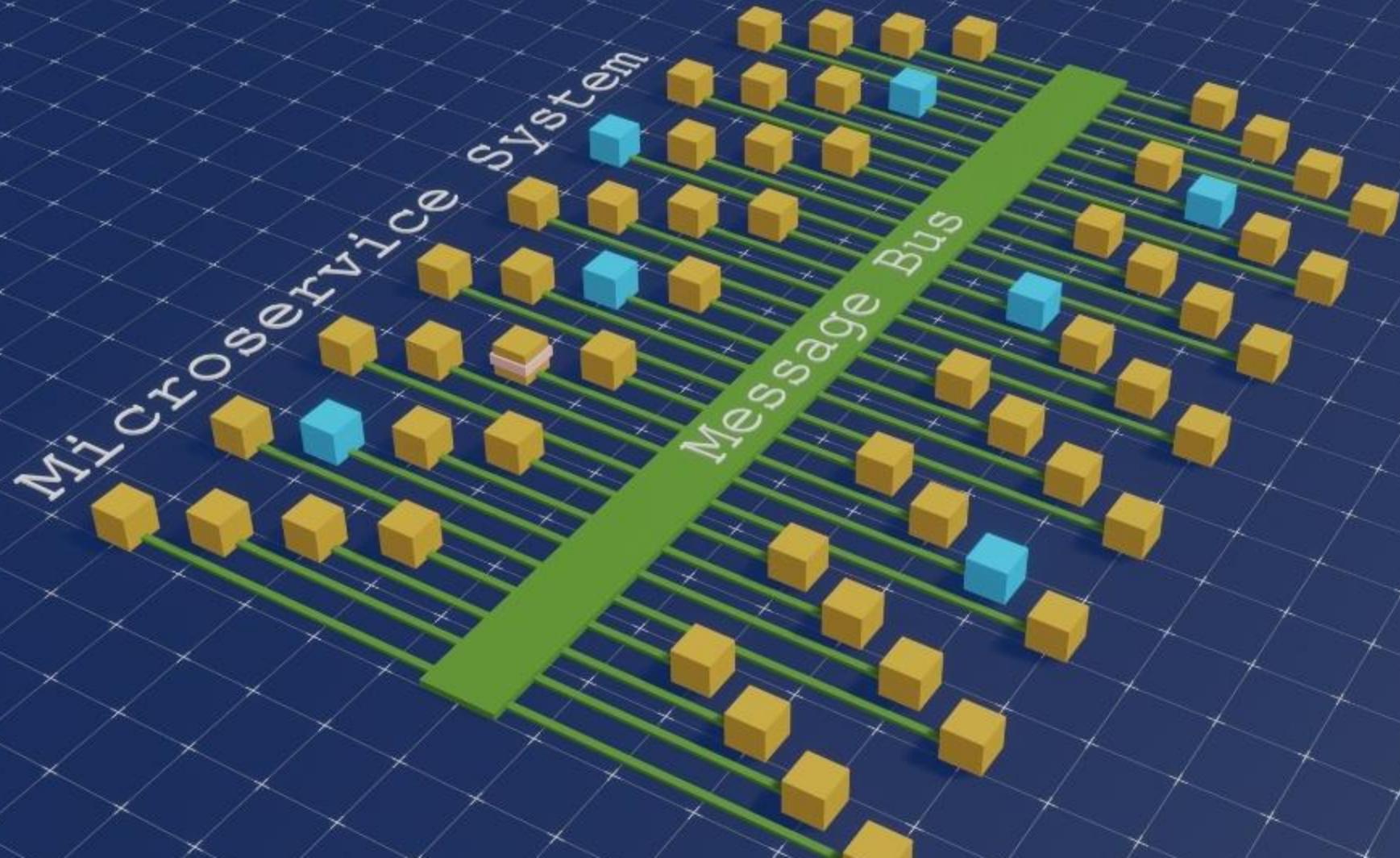


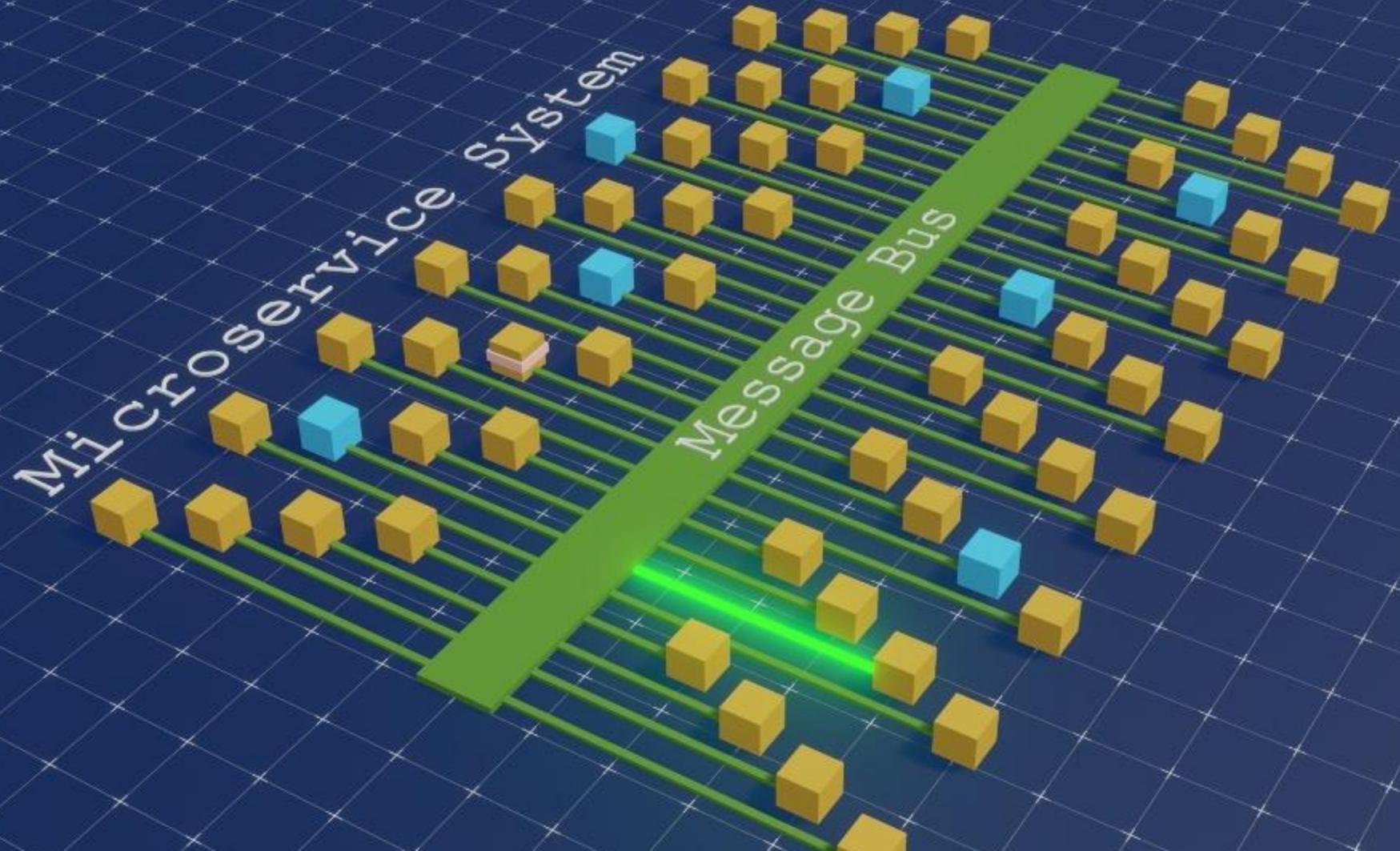


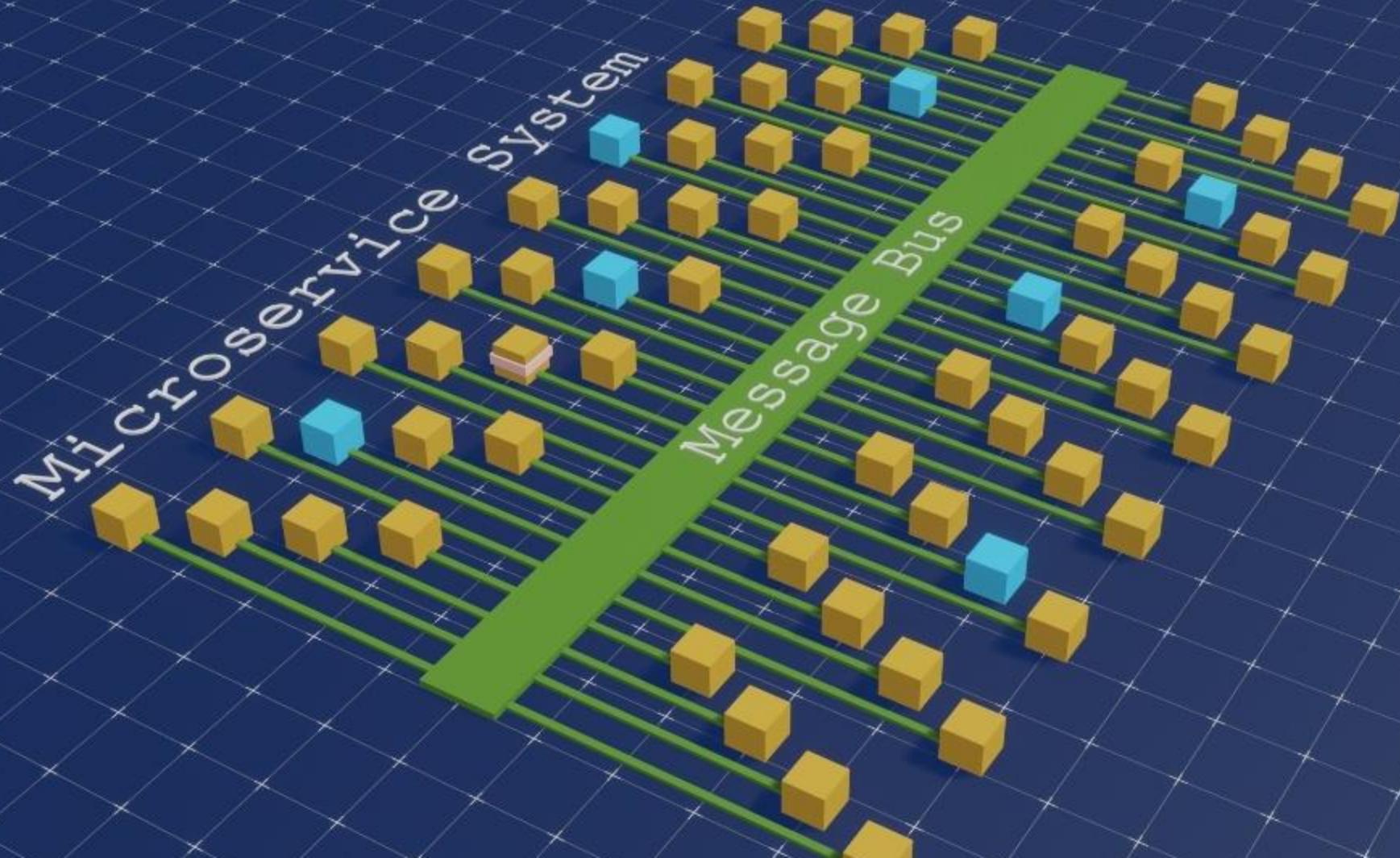


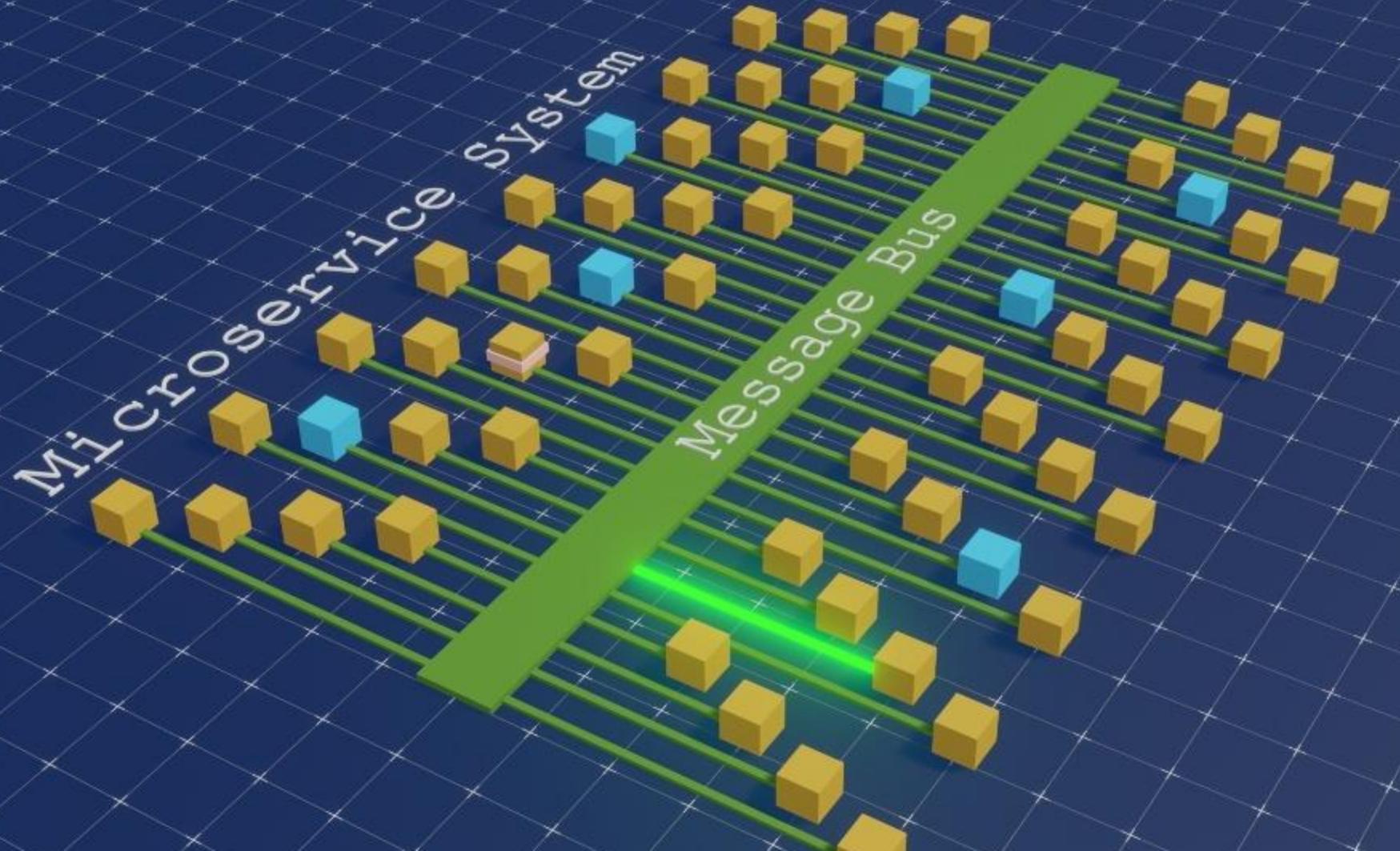


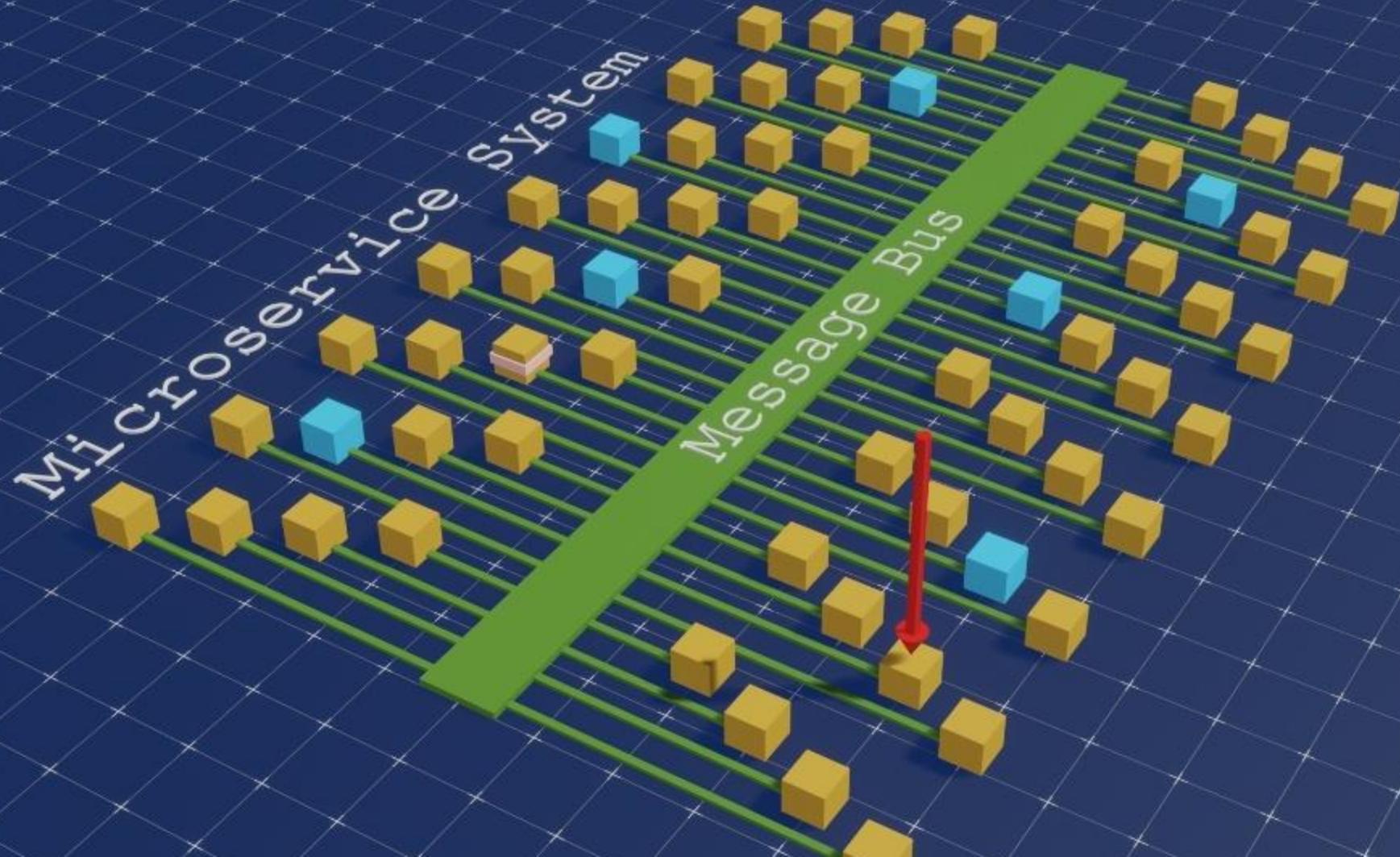








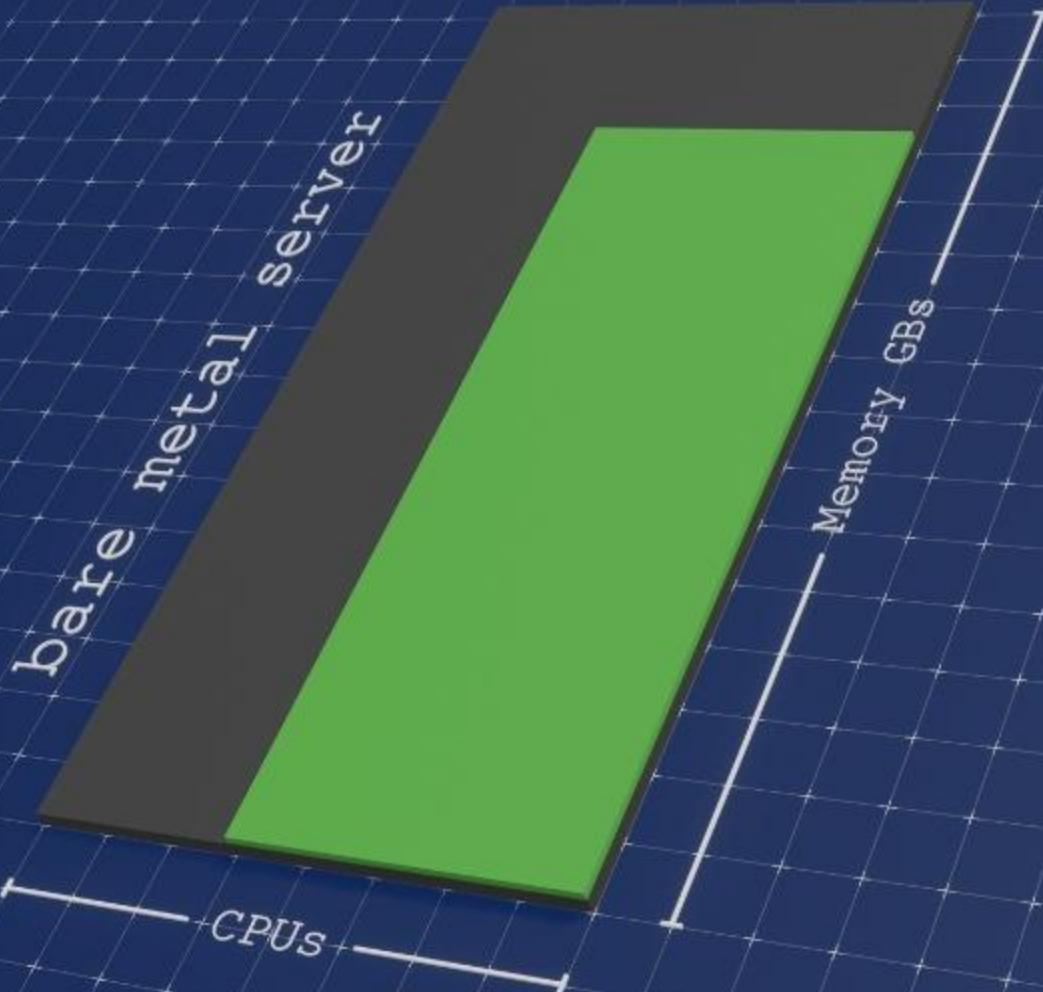




bare metal server

CPUs

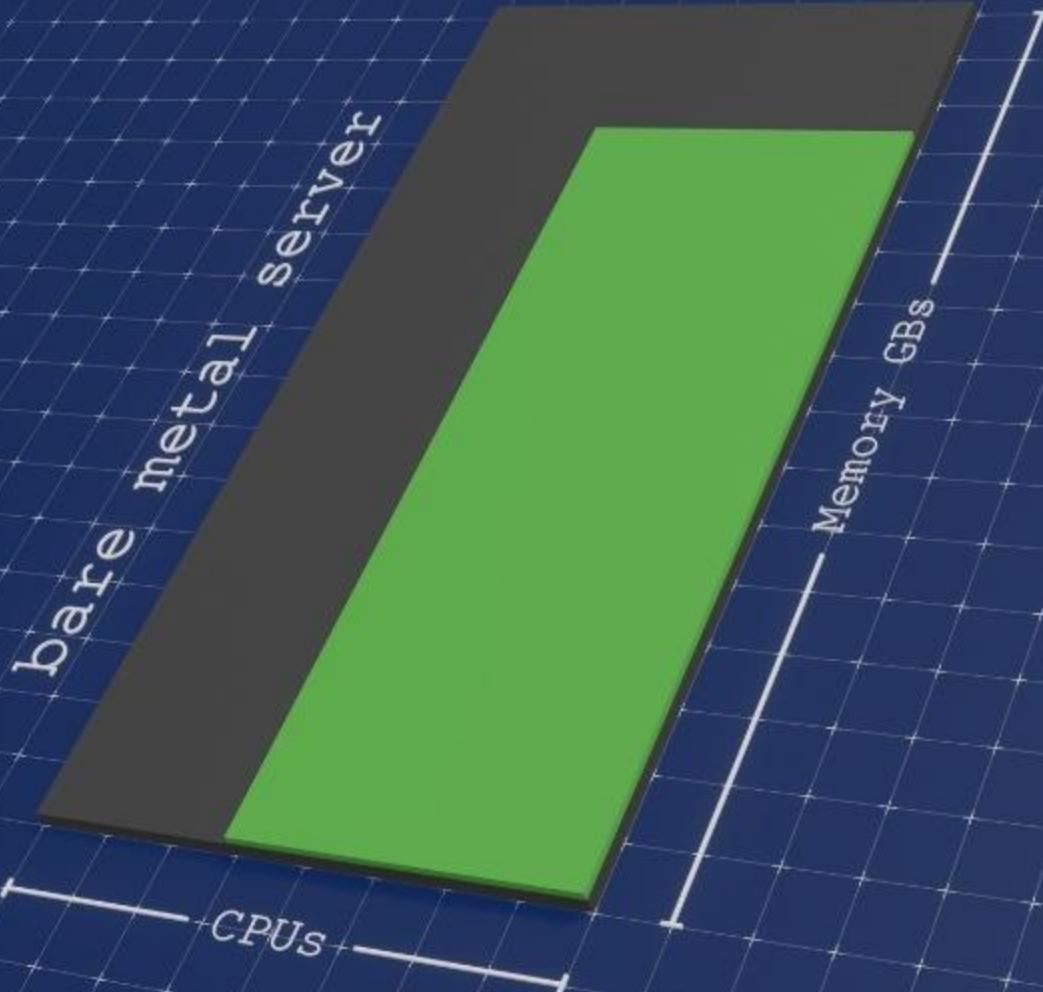
Memory GBs



bare metal server

CPUs

Memory GBs



bare metal server

CPUs

Memory GBs

bare metal server

CPUs

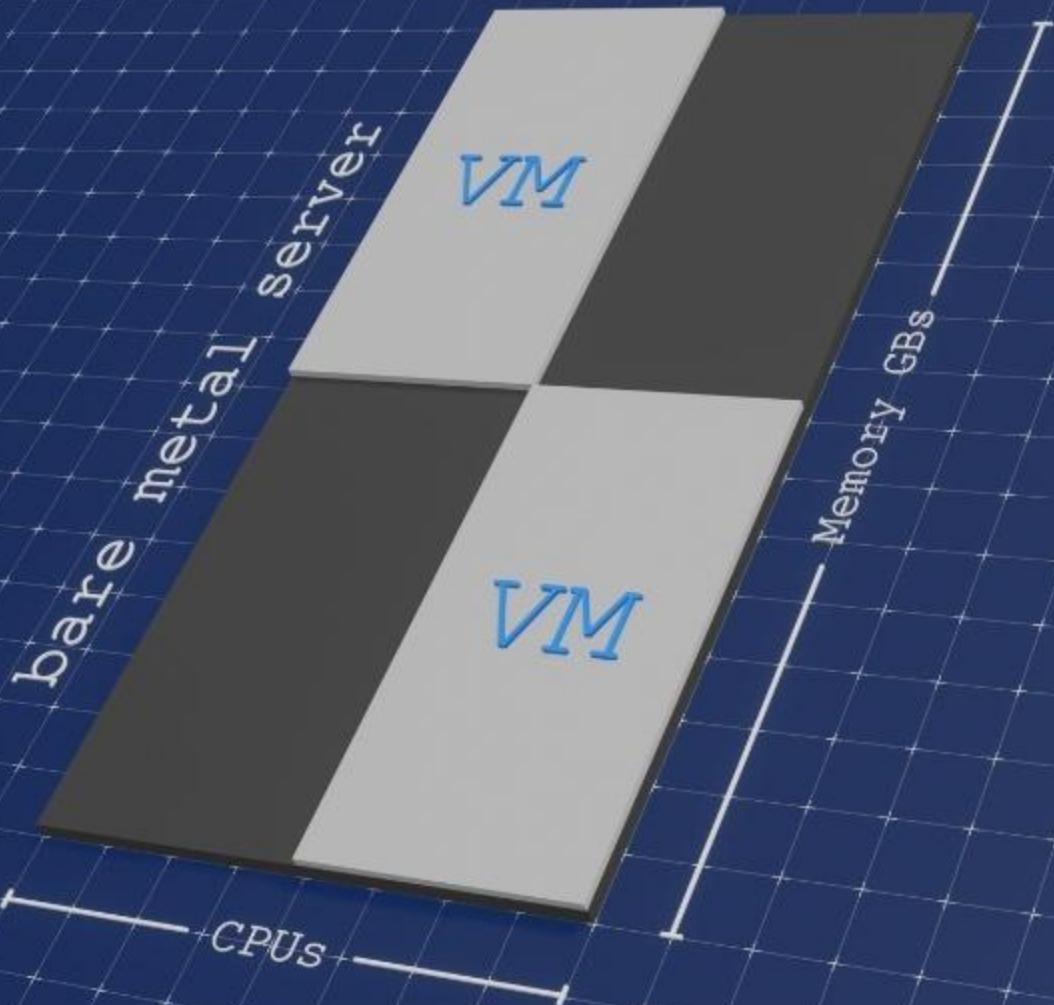
Memory GBs

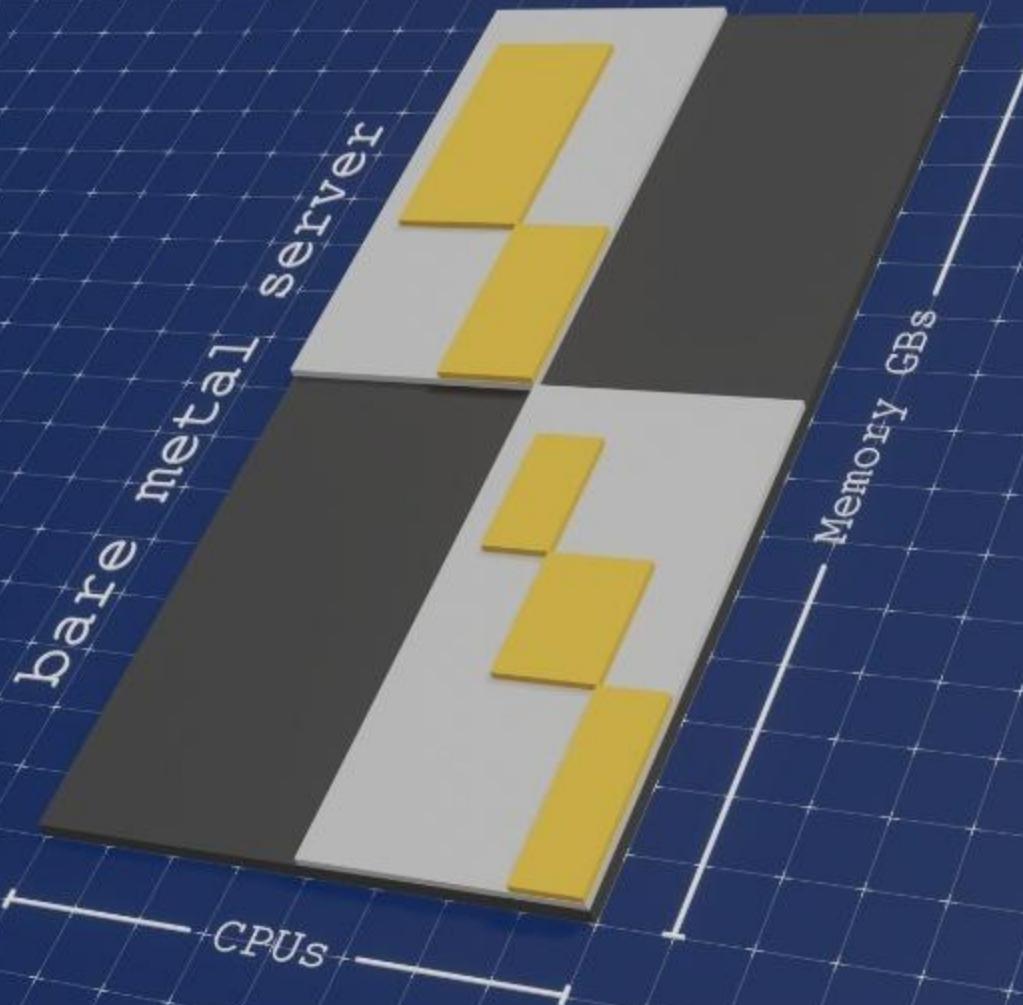
bare metal server

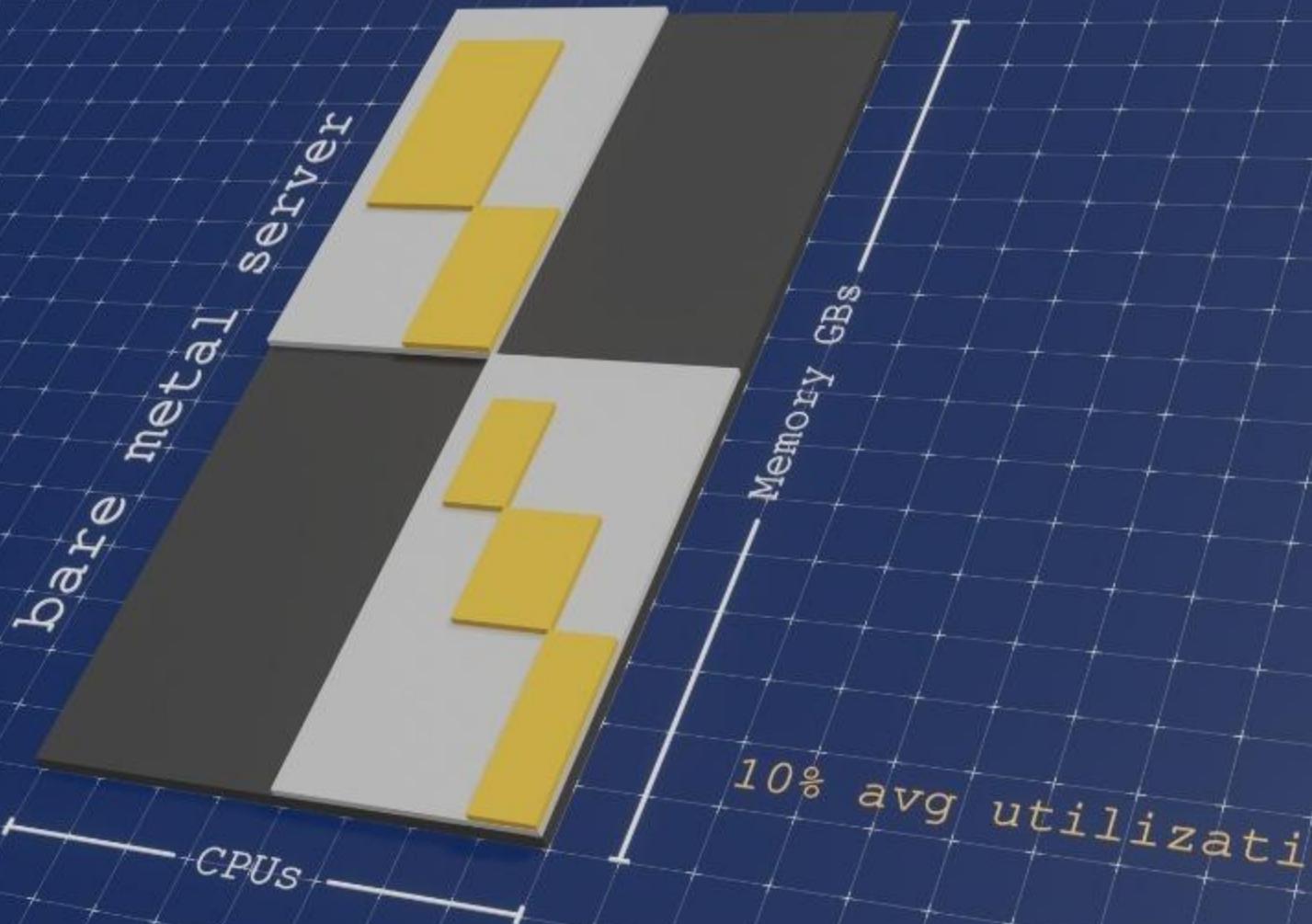
CPUs

Memory GBs

10% avg utilization





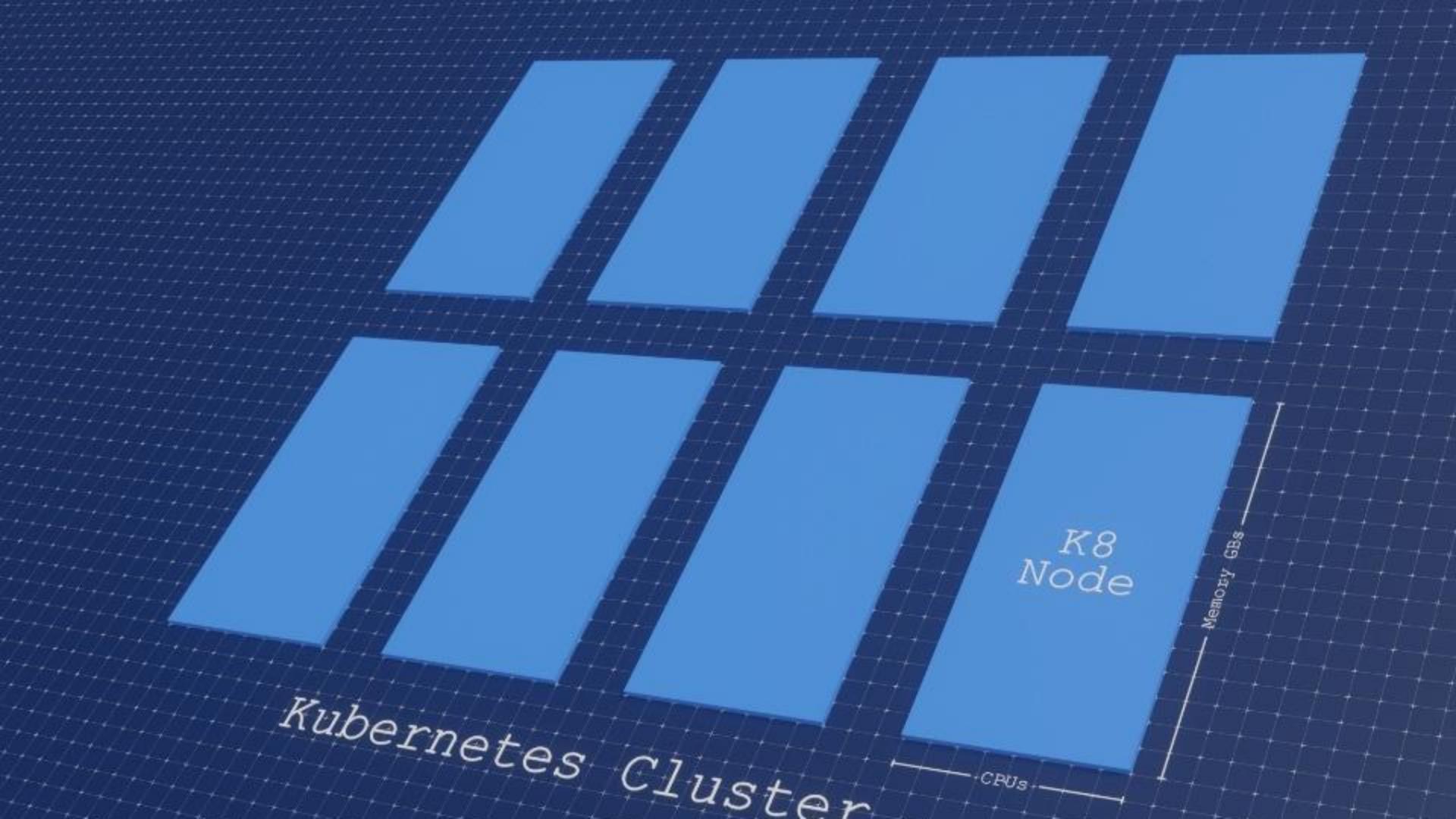


Kubernetes Cluster

*K8
Node*

CPUs

Memory GBs



K8 Node

Memory GB\$

Pod
Container

CPUs

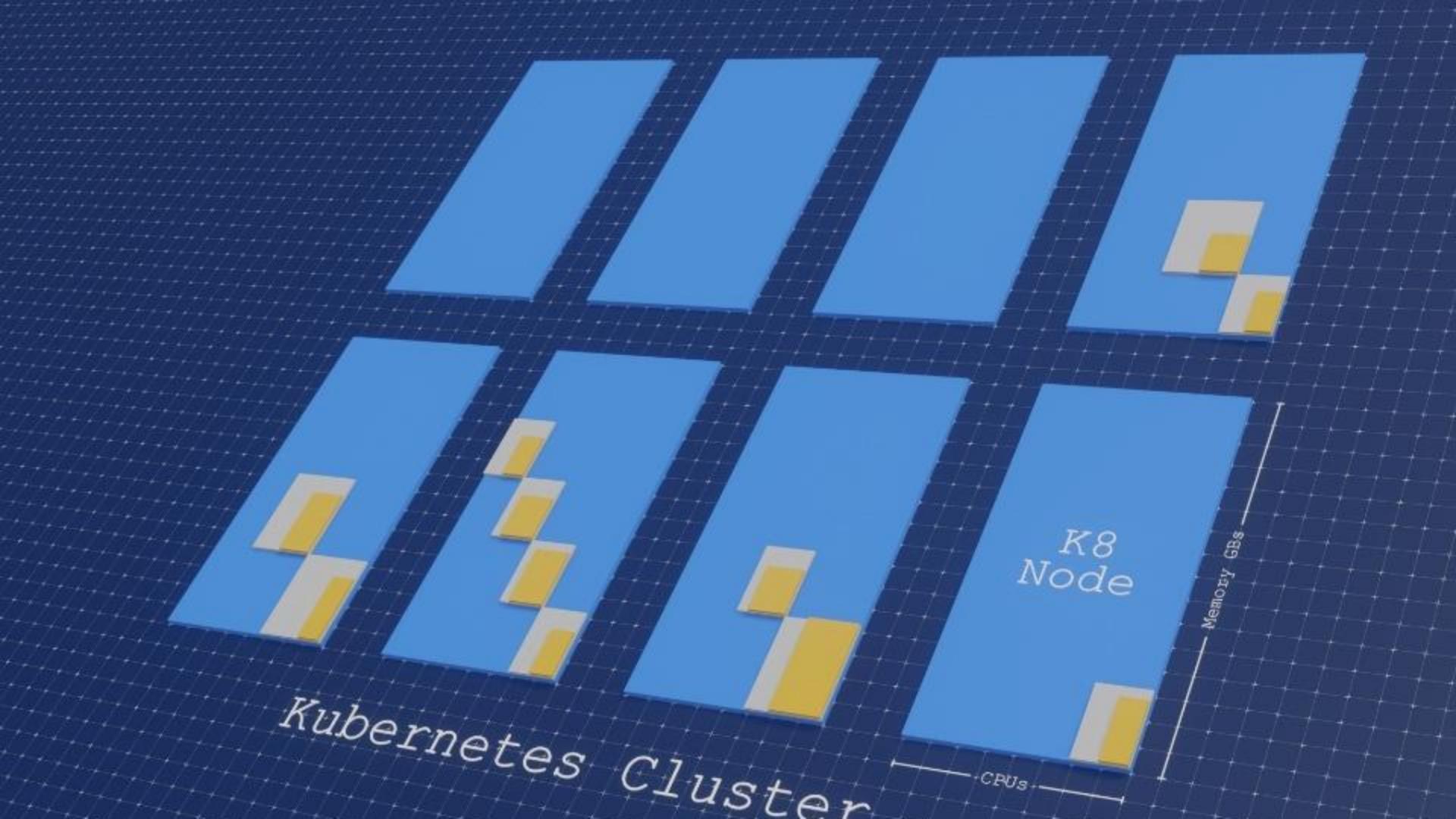
K8
Node

Pod
Container

CPUs

Memory GBs

Notes Cluster

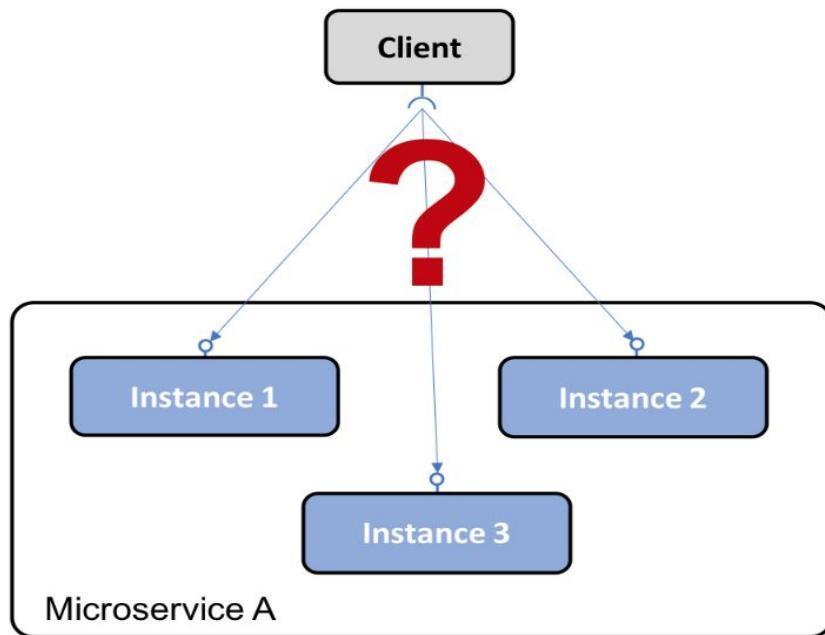


Design patterns for microservices

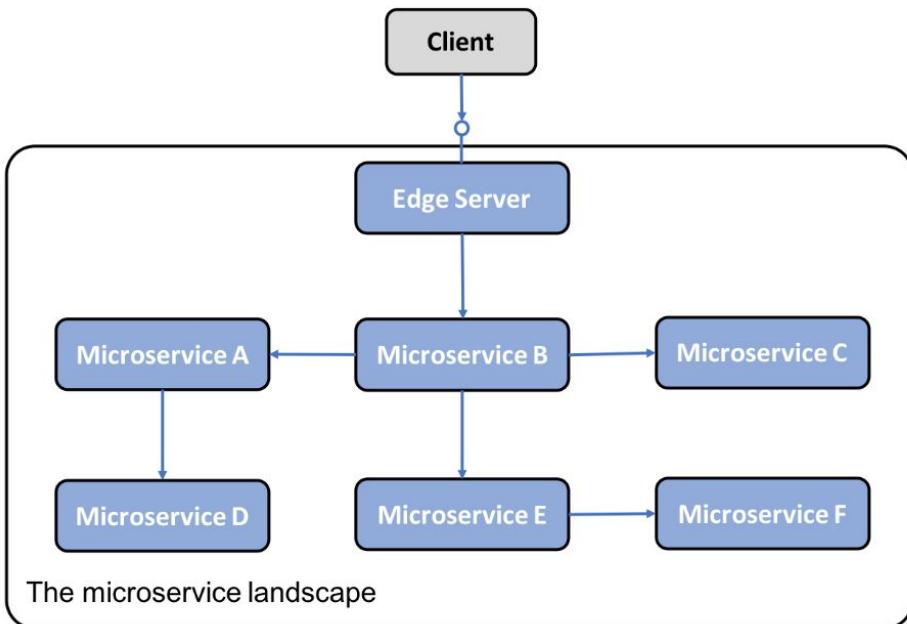
design patterns

- ▶ Service discovery
- ▶ Edge server
- ▶ Reactive microservices
- ▶ Central configuration
- ▶ Centralized log analysis
- ▶ Distributed tracing
- ▶ Circuit Breaker
- ▶ Control loop
- ▶ Centralized monitoring and alarms

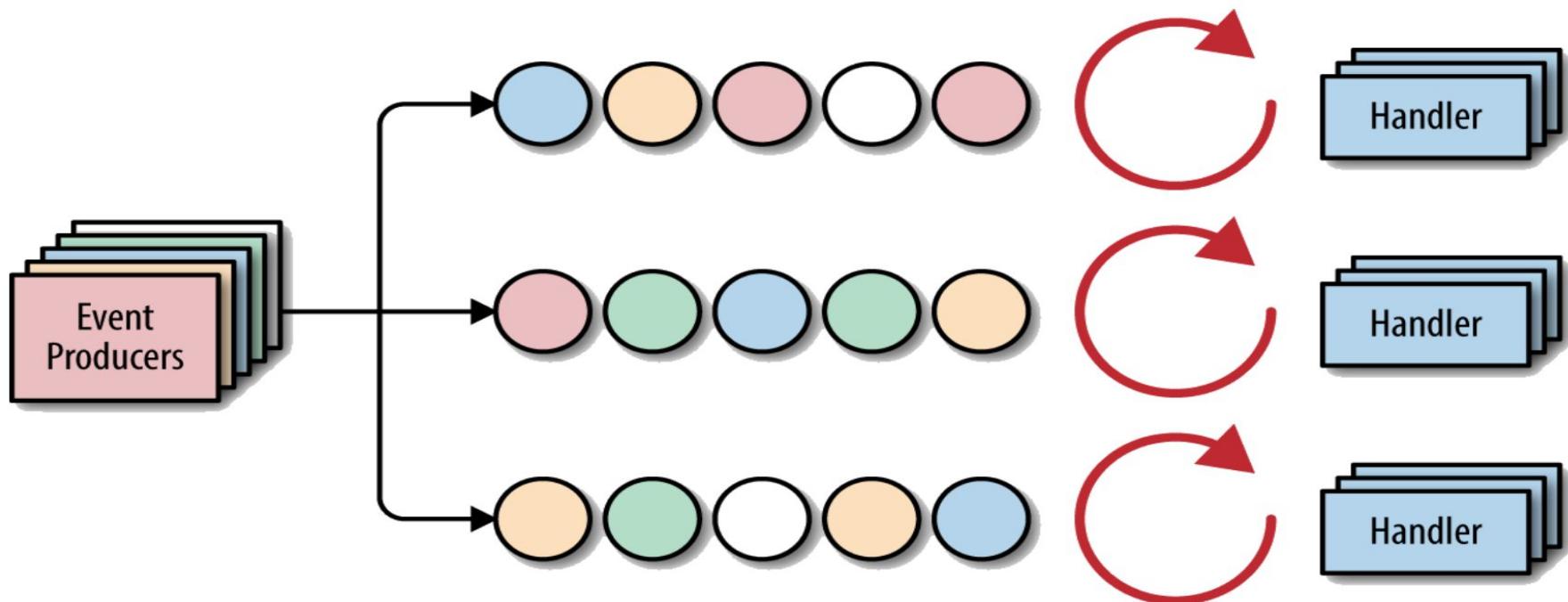
1. Service Discovery



2. Edge (Gateway/Load Balancer) Server



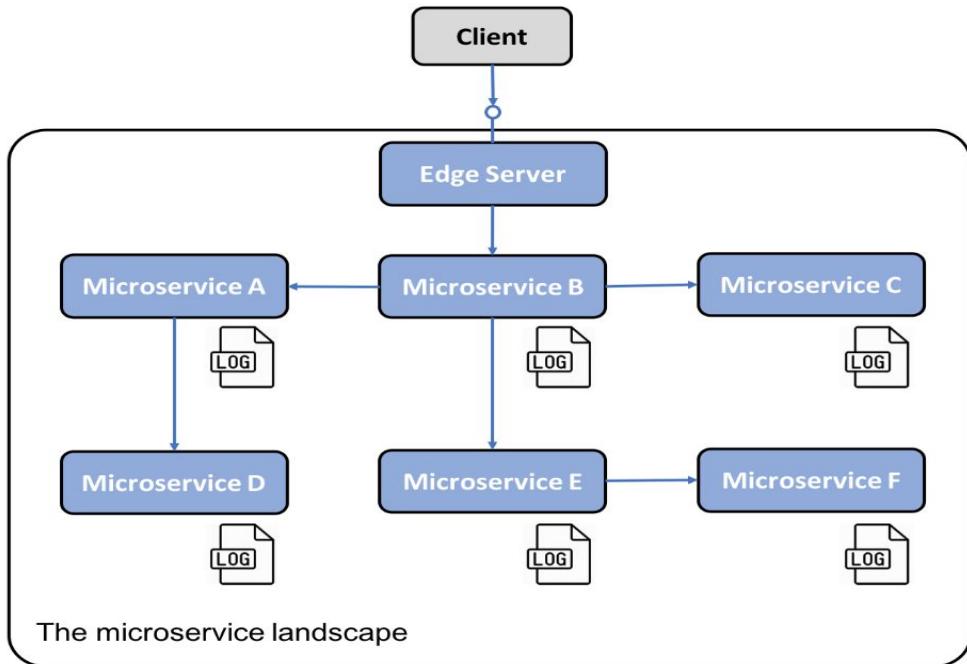
3. Reactive Microservices



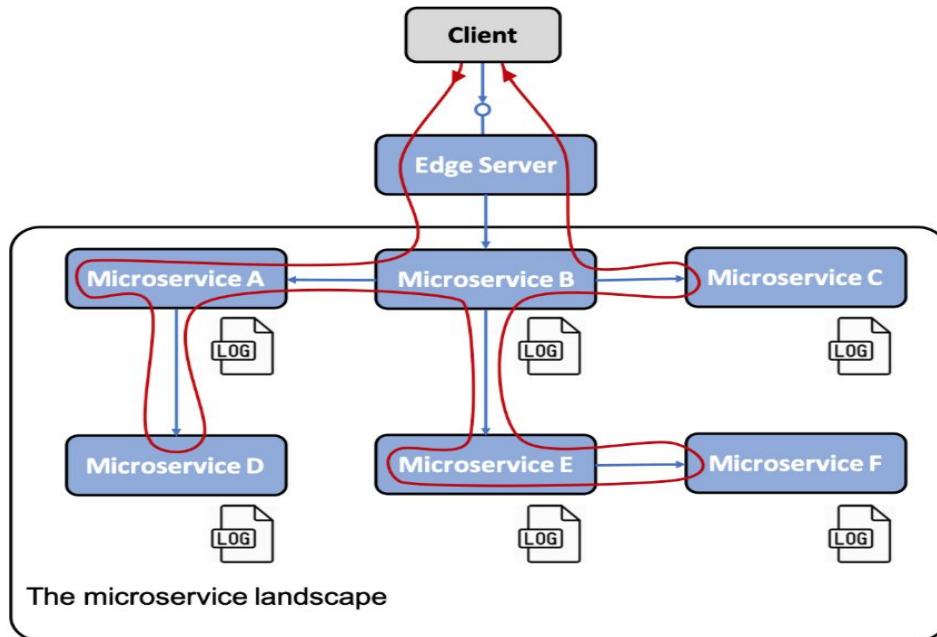
4. Central Configuration

Add a new component,
a configuration server, to the system landscape to store the configuration of all the microservices.

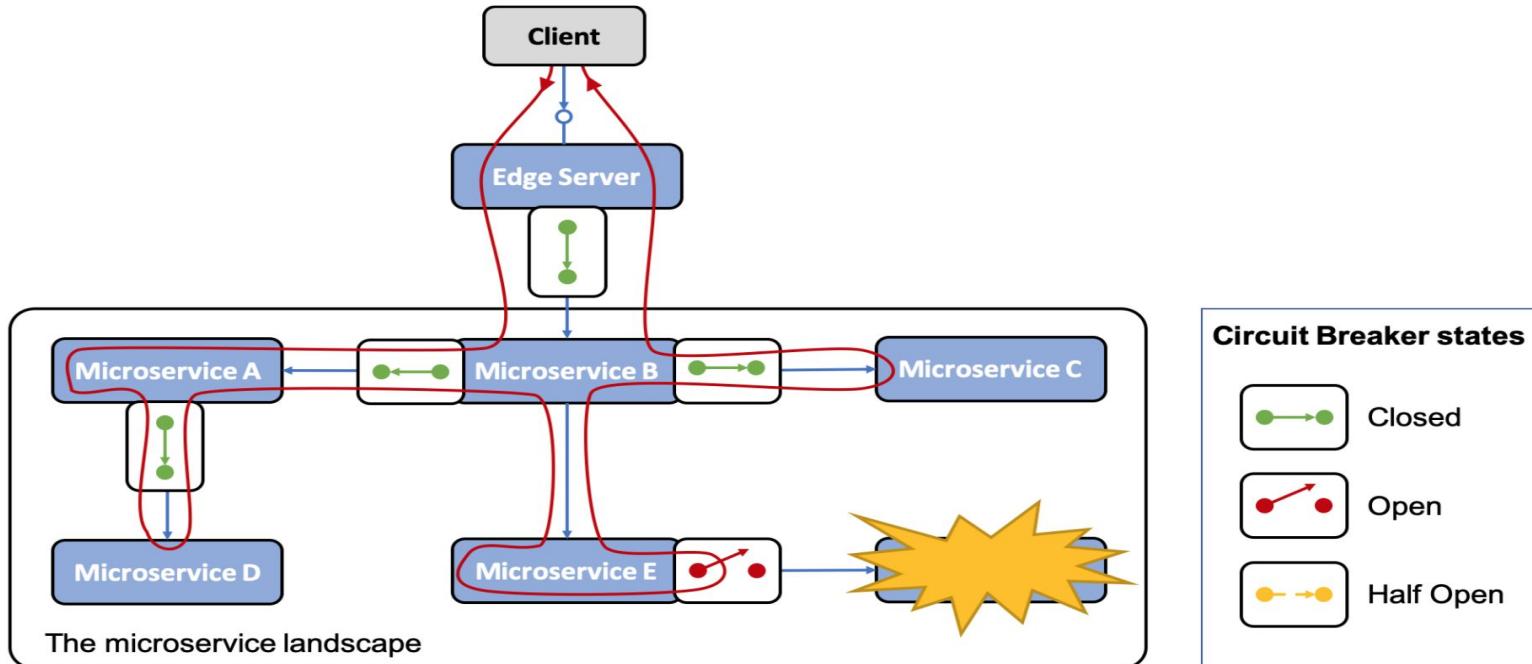
5. Centralized Log Analysis



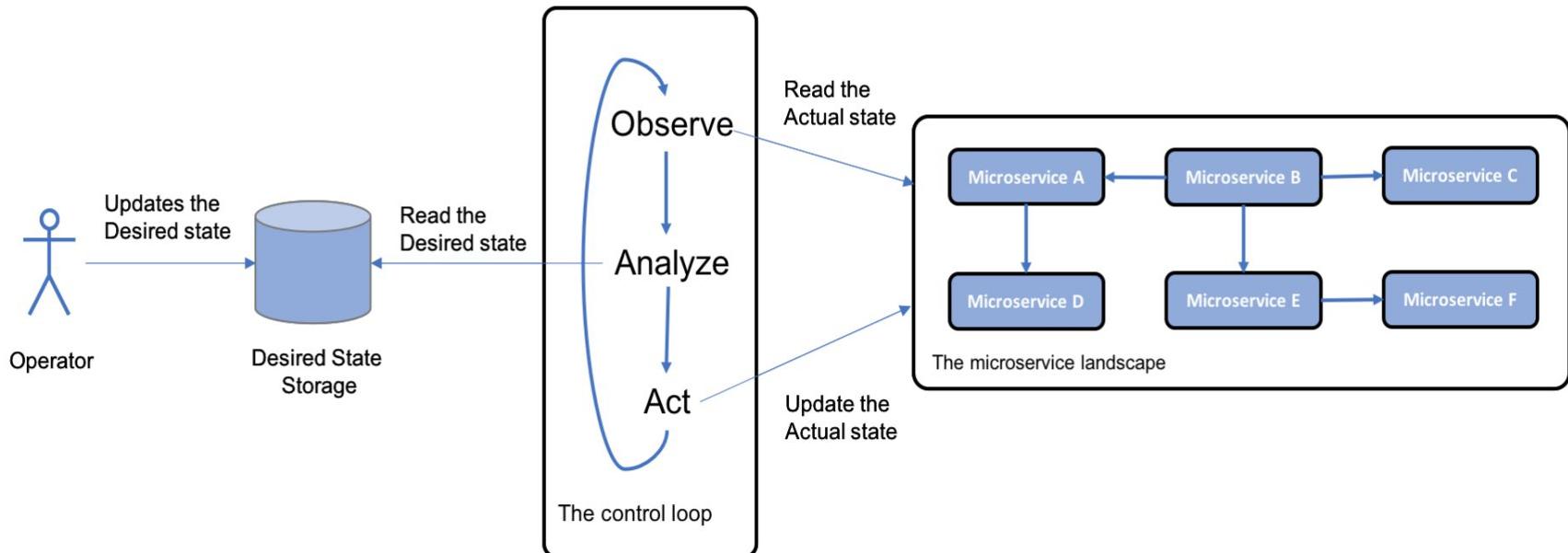
6. Distributed Tracing



7. Circuit Breaker



8. Control Loop



9. Centralized monitoring & alarms

