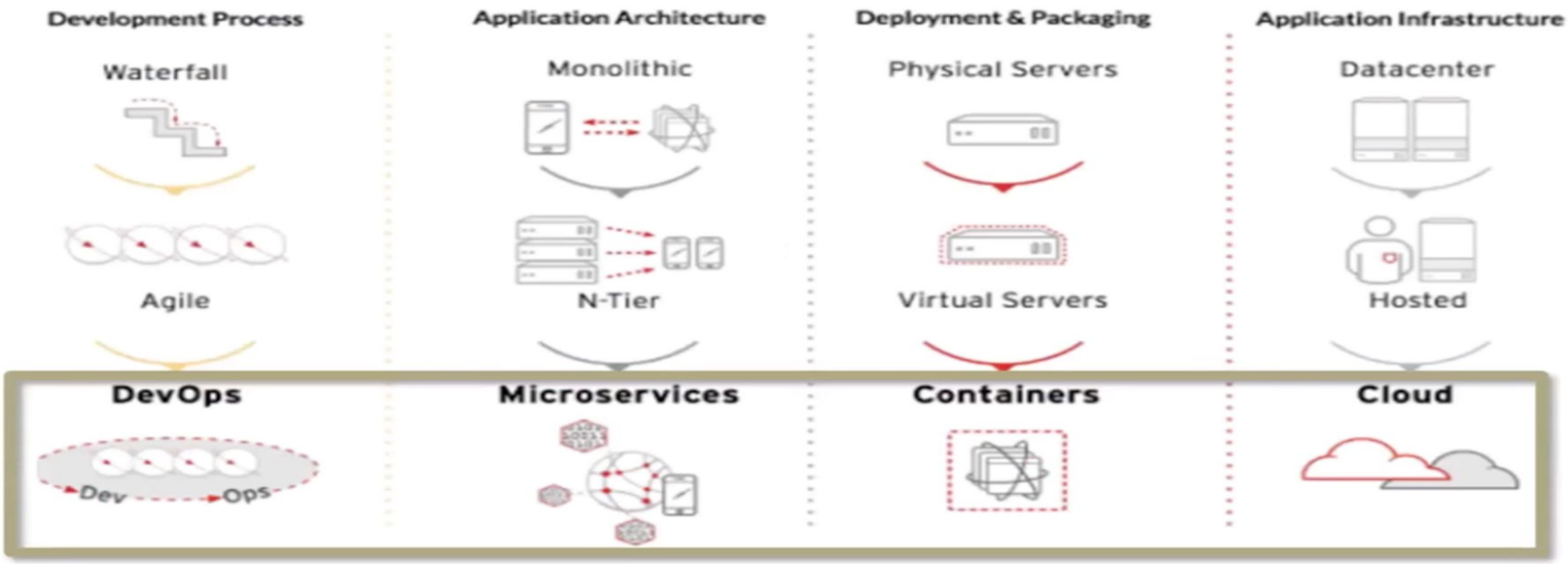
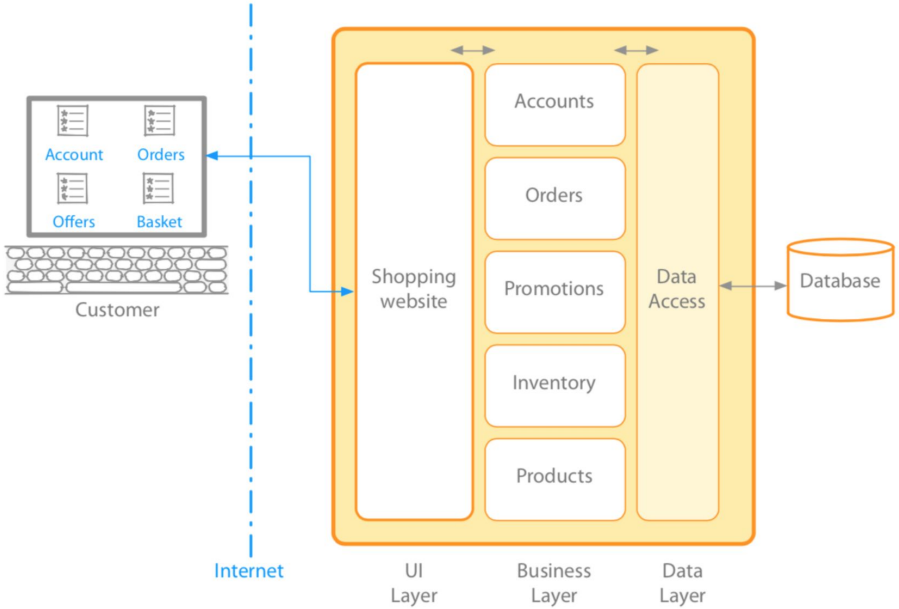
Microservices

Software Engineering evolution



Monolithic architecture - Ex



Monolithic architecture - pros

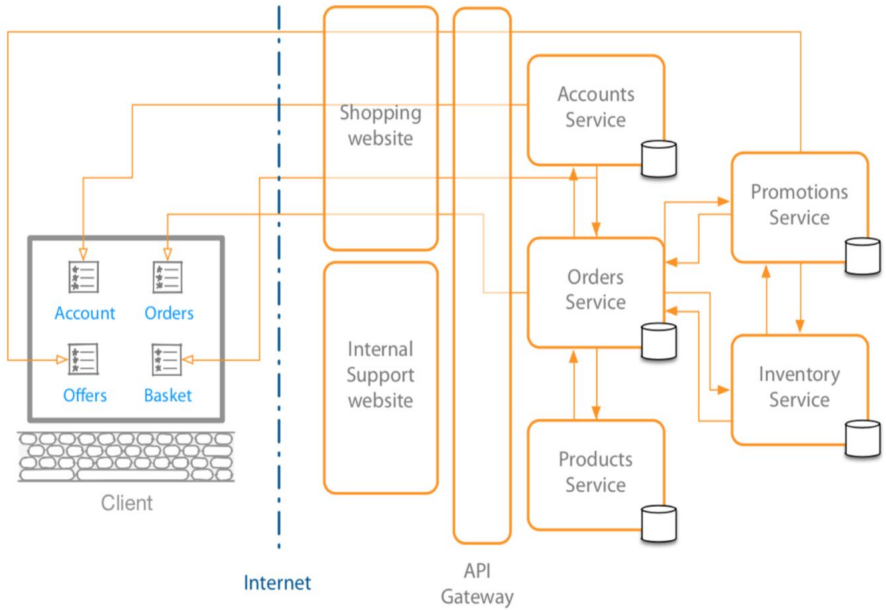
- Simpler development & deployment
- Fewer cross-cutting concerns
- Better performance

Monolithic architecture - cons

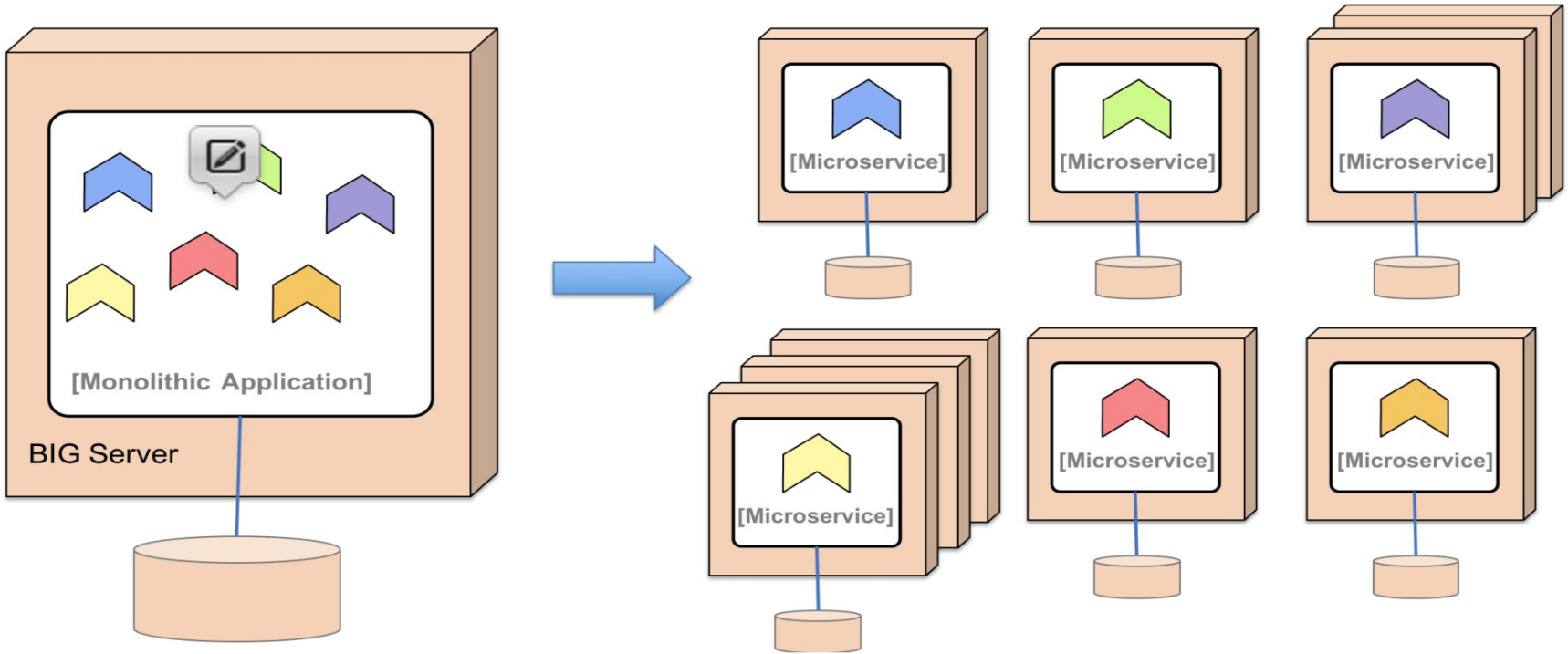
- Larger codebase
- Difficult to adopt new technologies
- Failure could affect whole system
- Scaling requires duplication of the whole
- Limited agility
  - every small update requires a full redeployment
  - all developers have to wait until it’s done
  - When several teams are working on the same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

Enter Microservices

Microservice - Ex



Defining a microservice

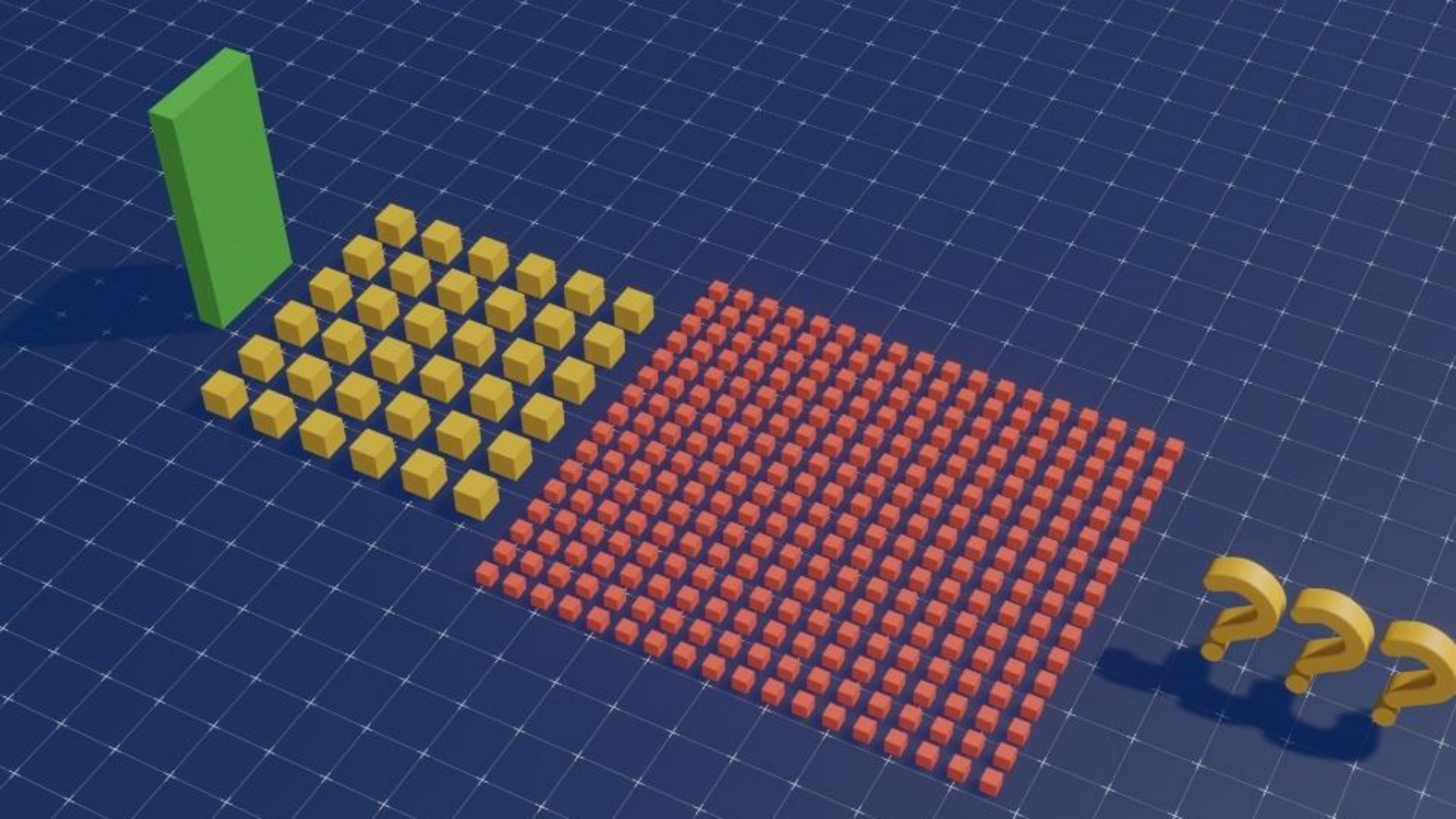


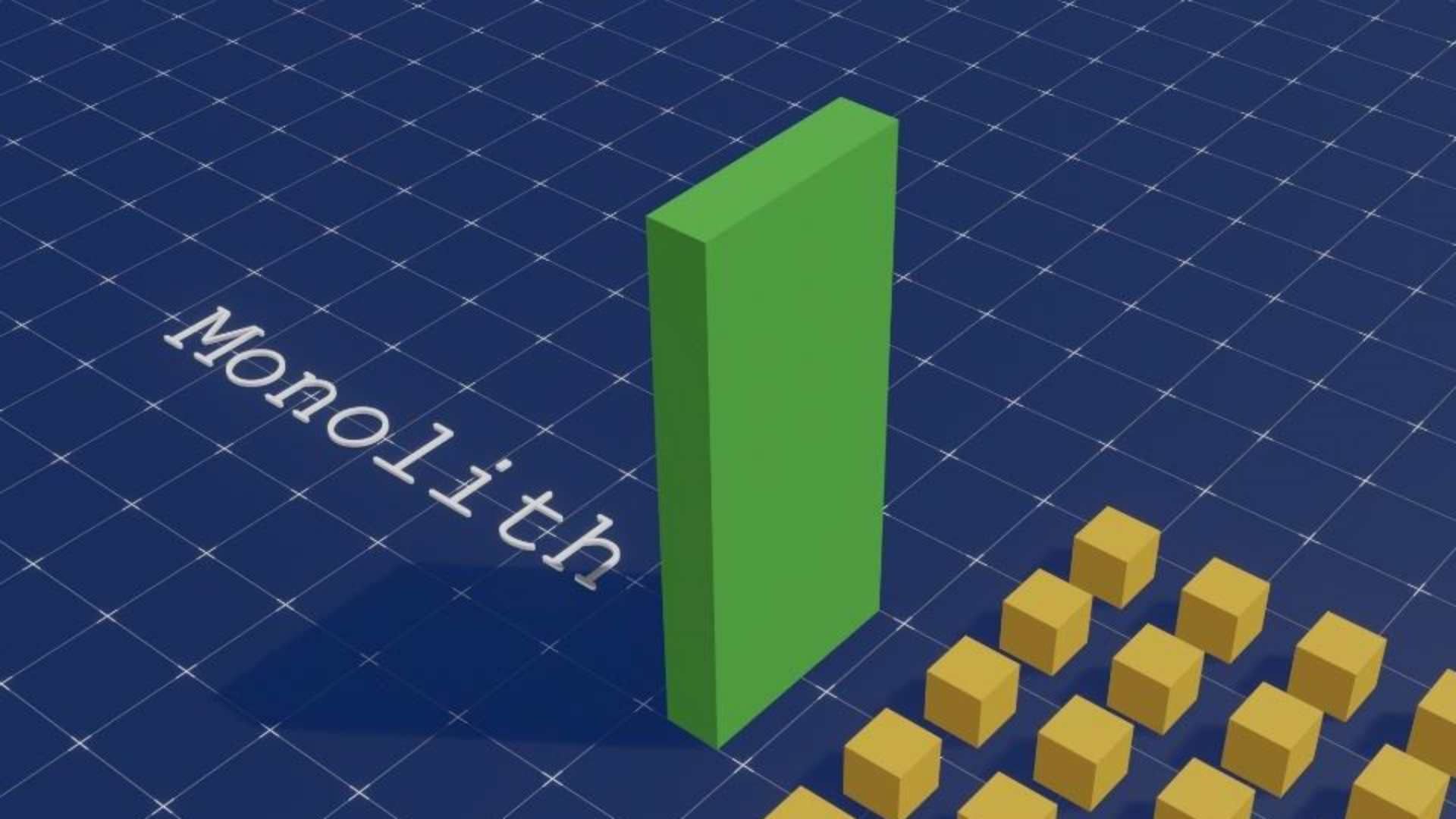
Microservice - characteristics

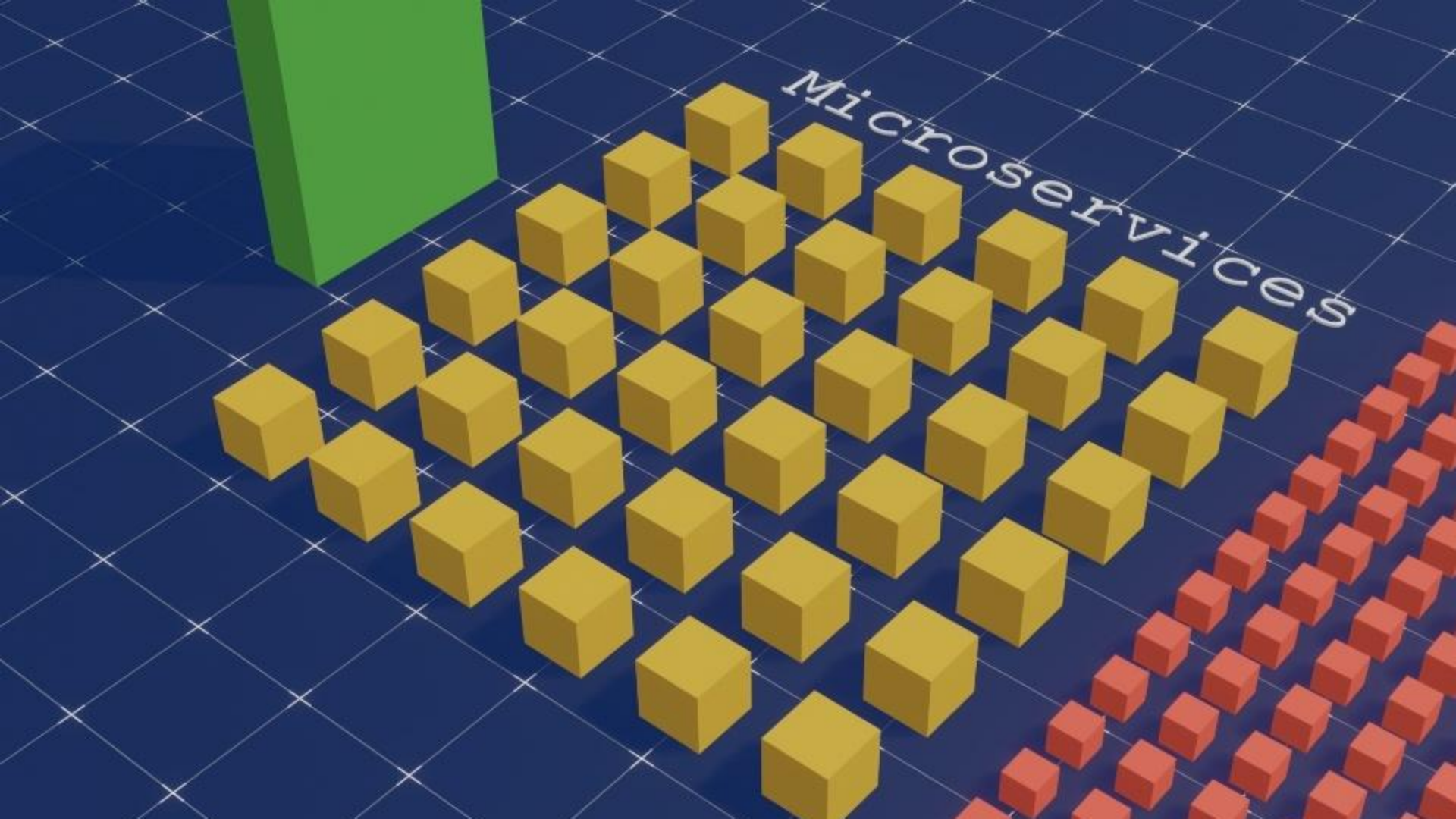
- Must conforms to a shared-nothing architecture
- Must only communicate with well defined interfaces (Api calls | Messages)
- Must be deployed as separate runtime process e.g Docker
- Microservice instances are stateless

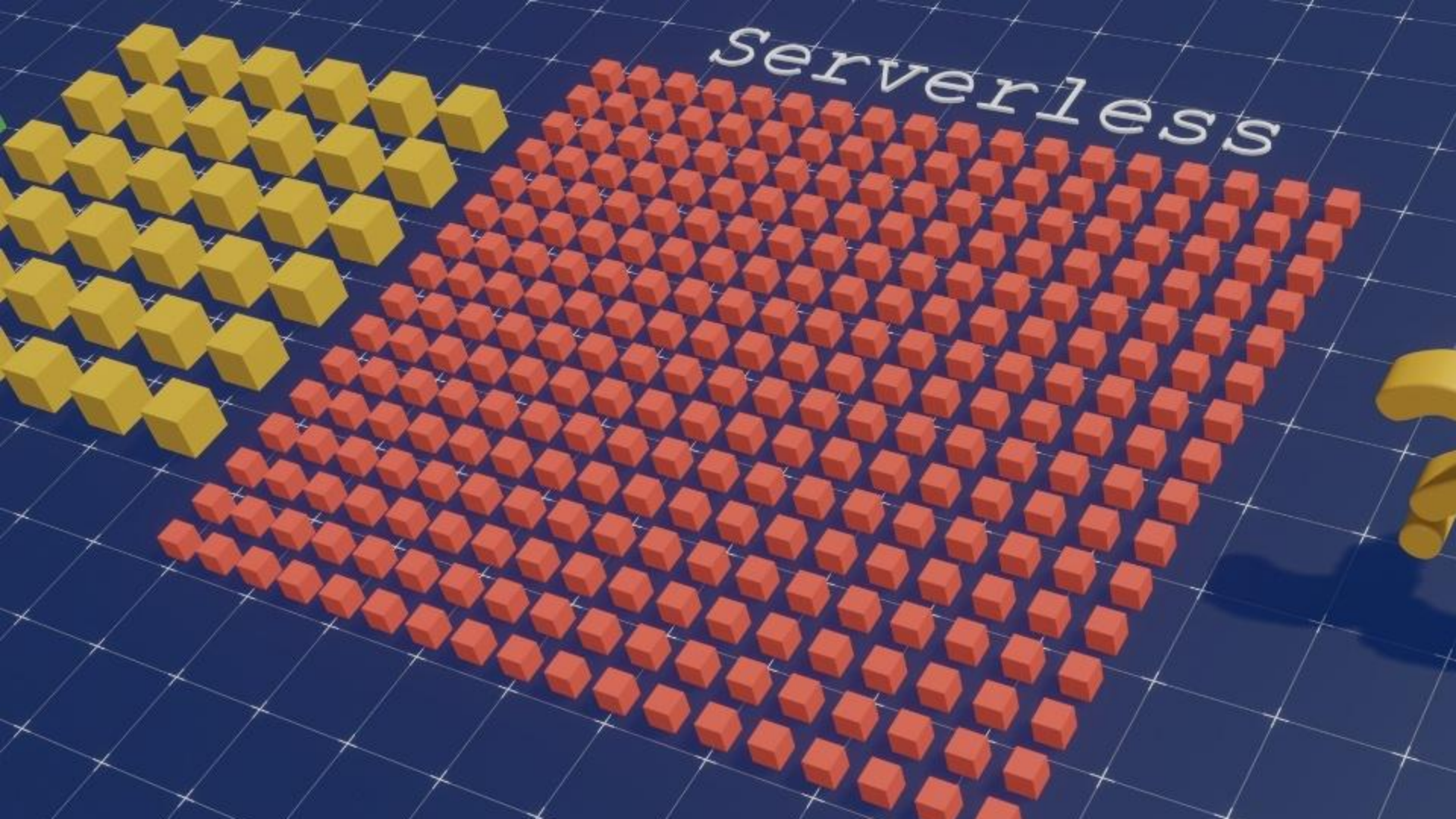
Microservice - benefits

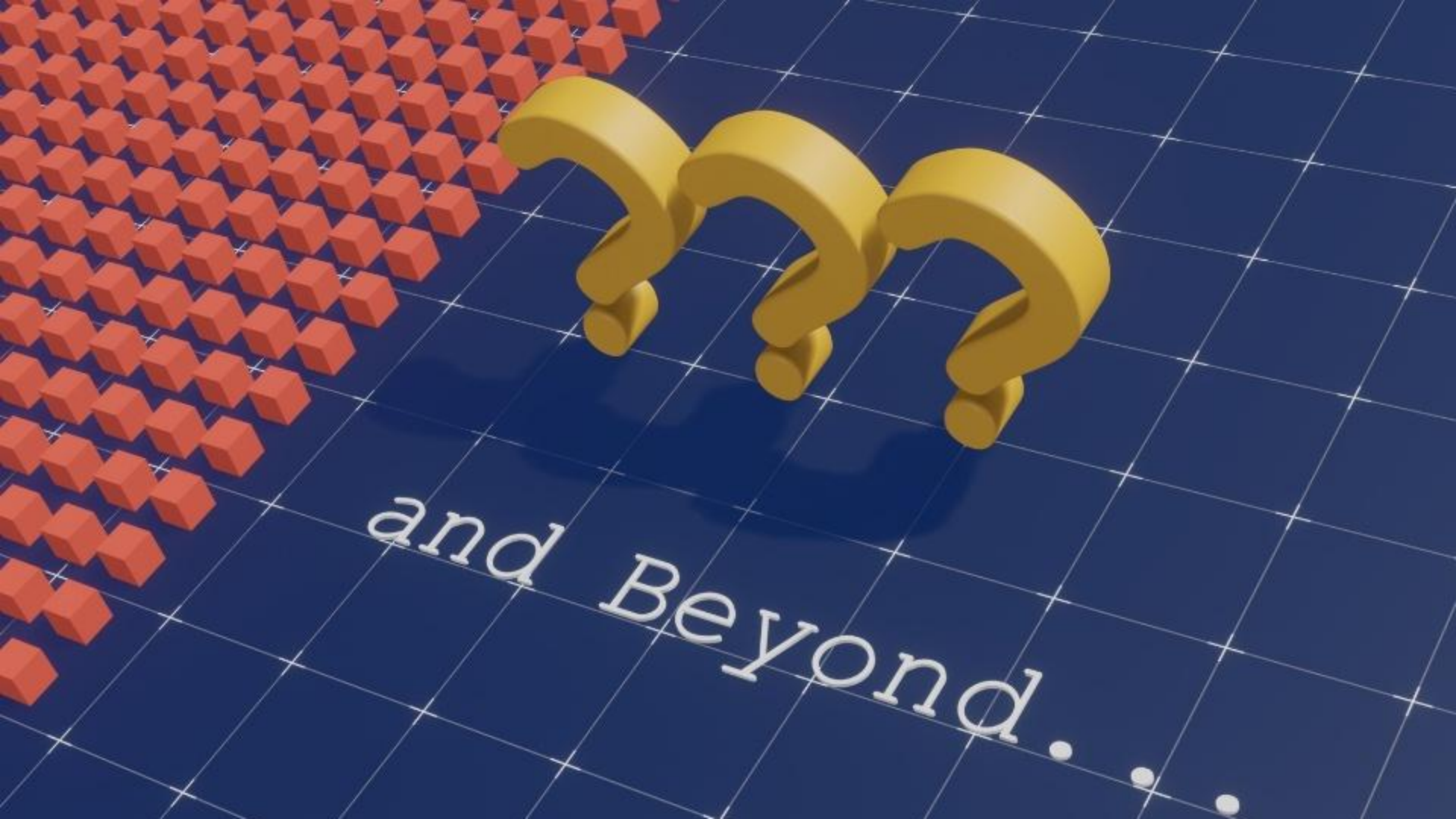
- Easy to develop,test and deploy
- Increased agility
- Ability to **scale** horizontally

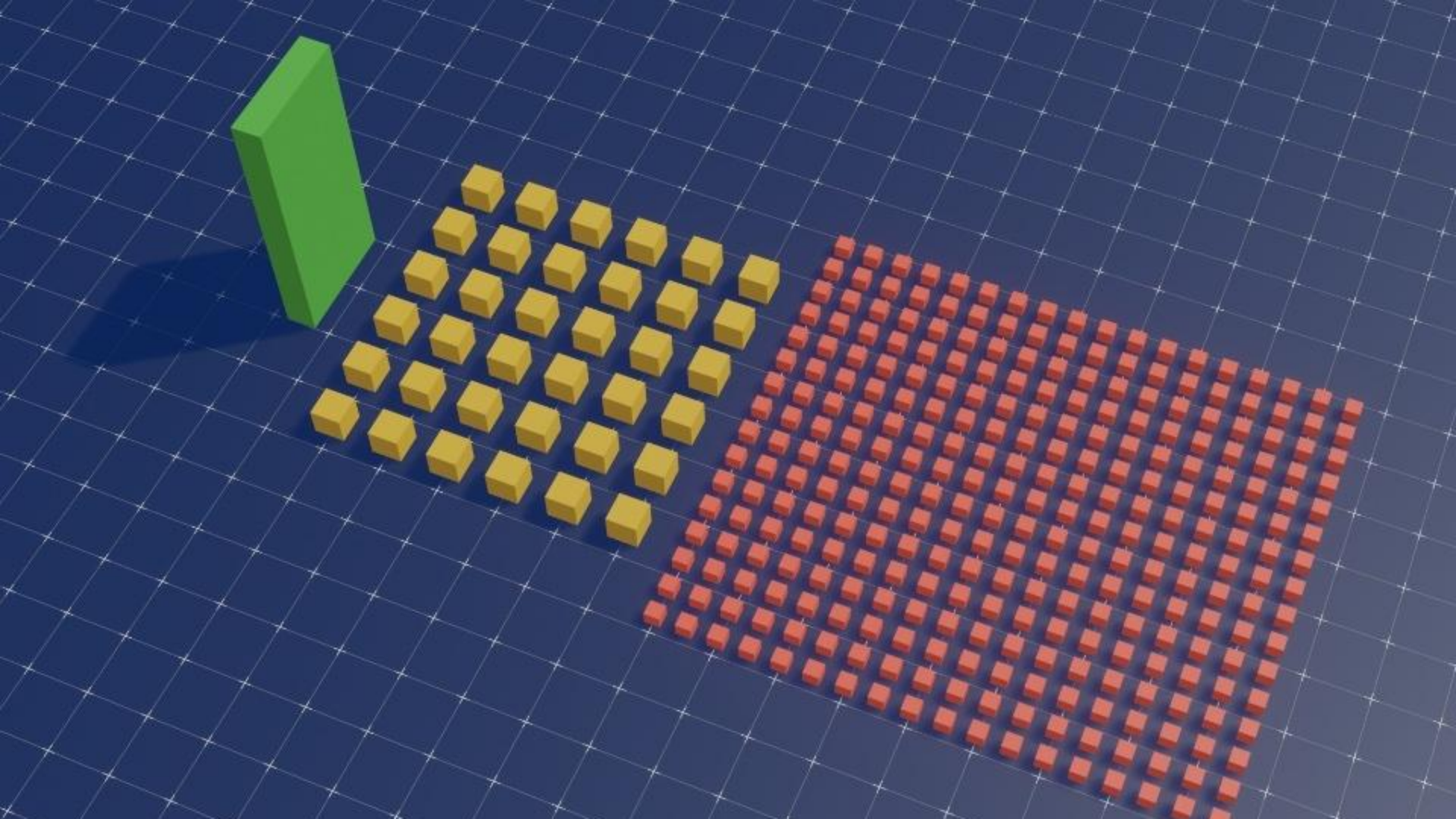


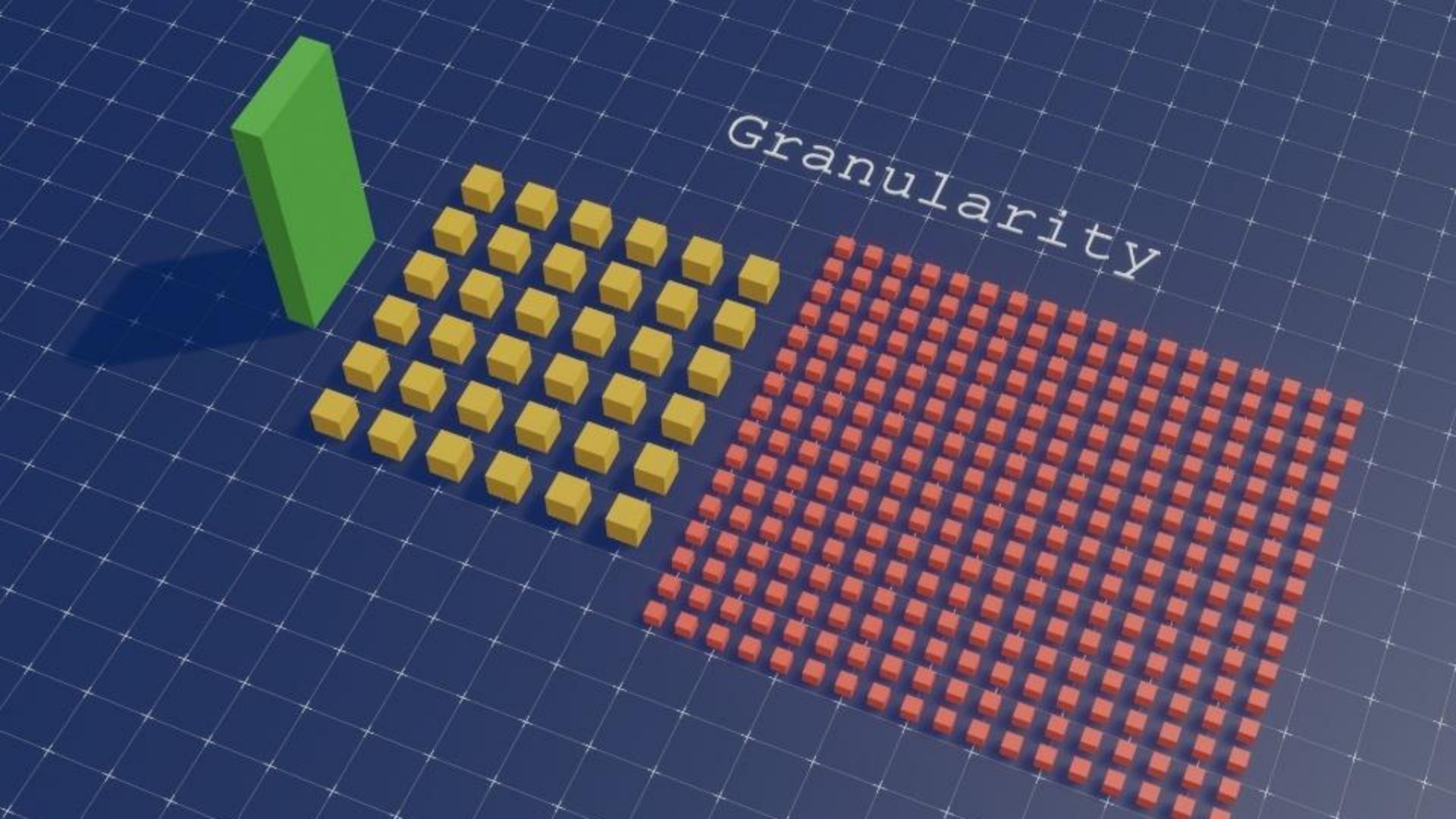

Monolith

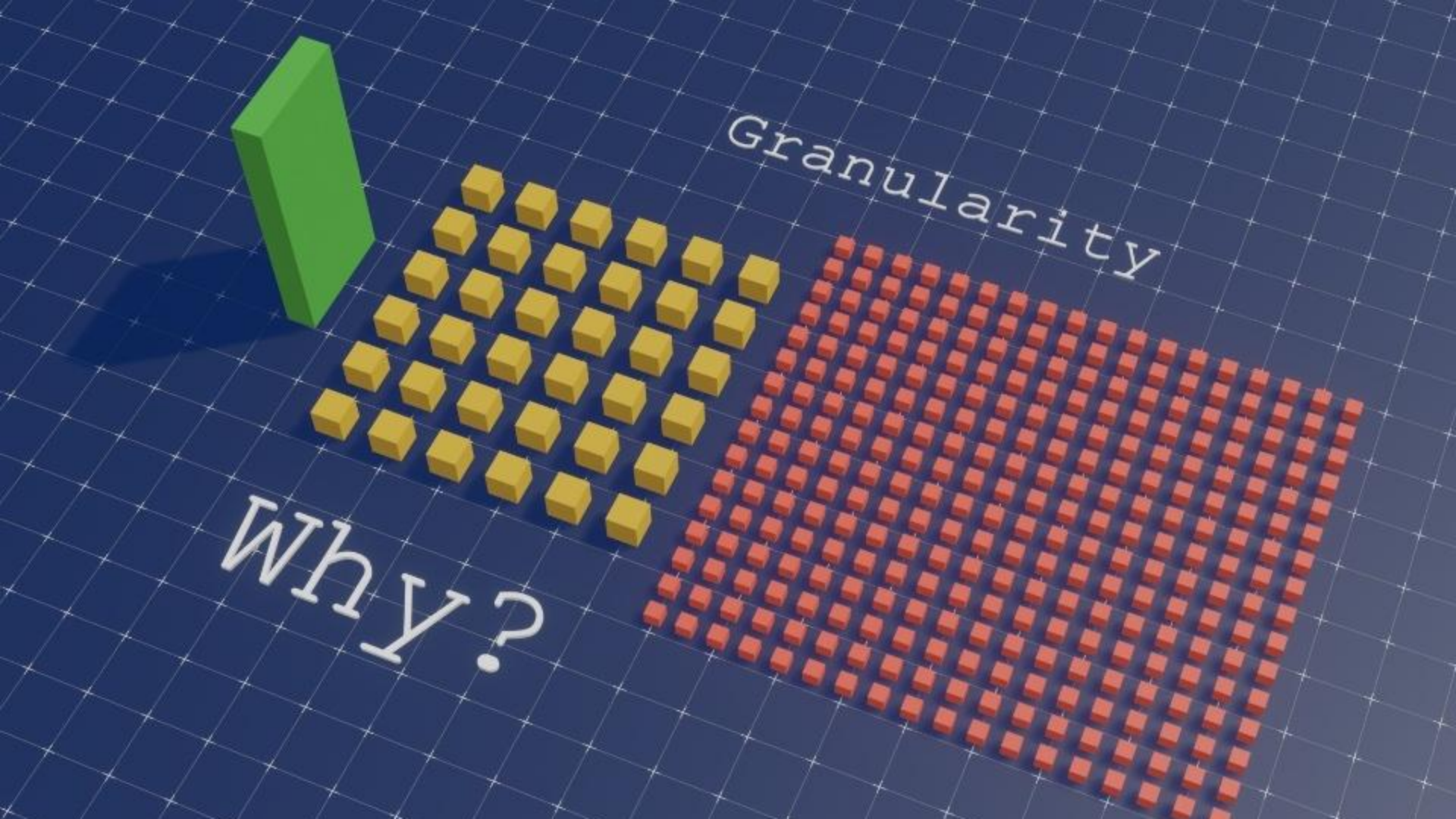

Microservices

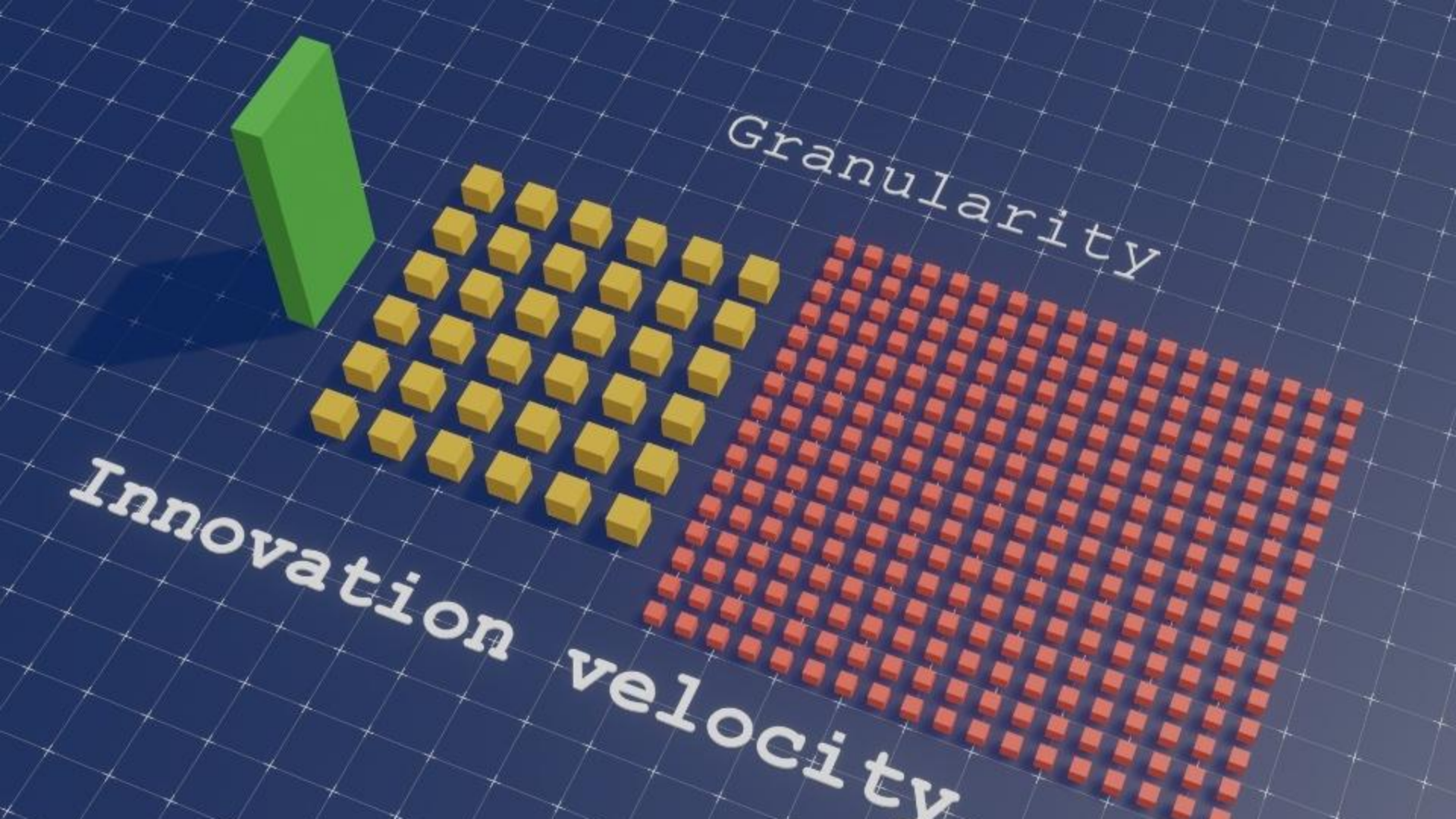

Serverless

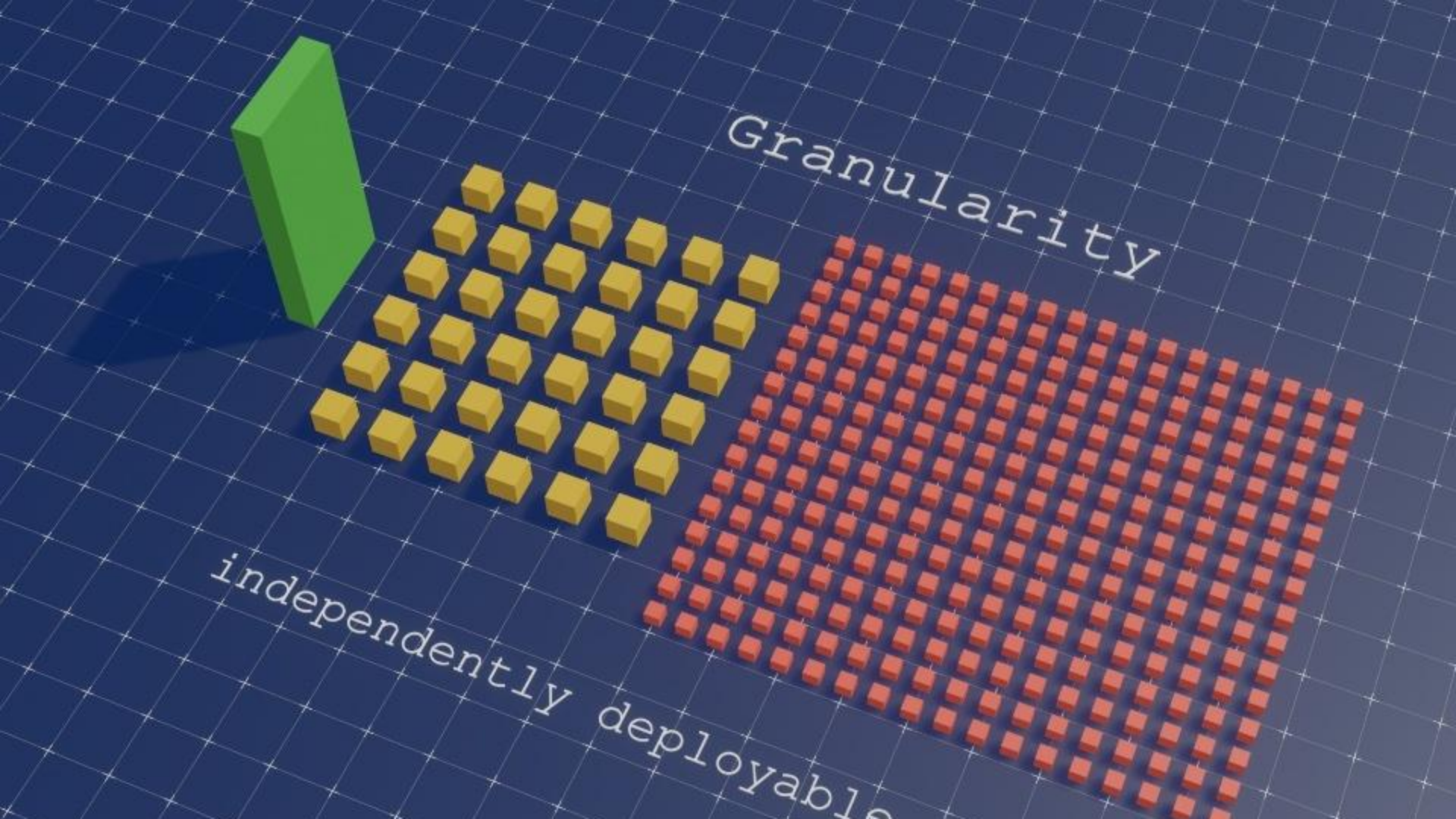

and Beyond...

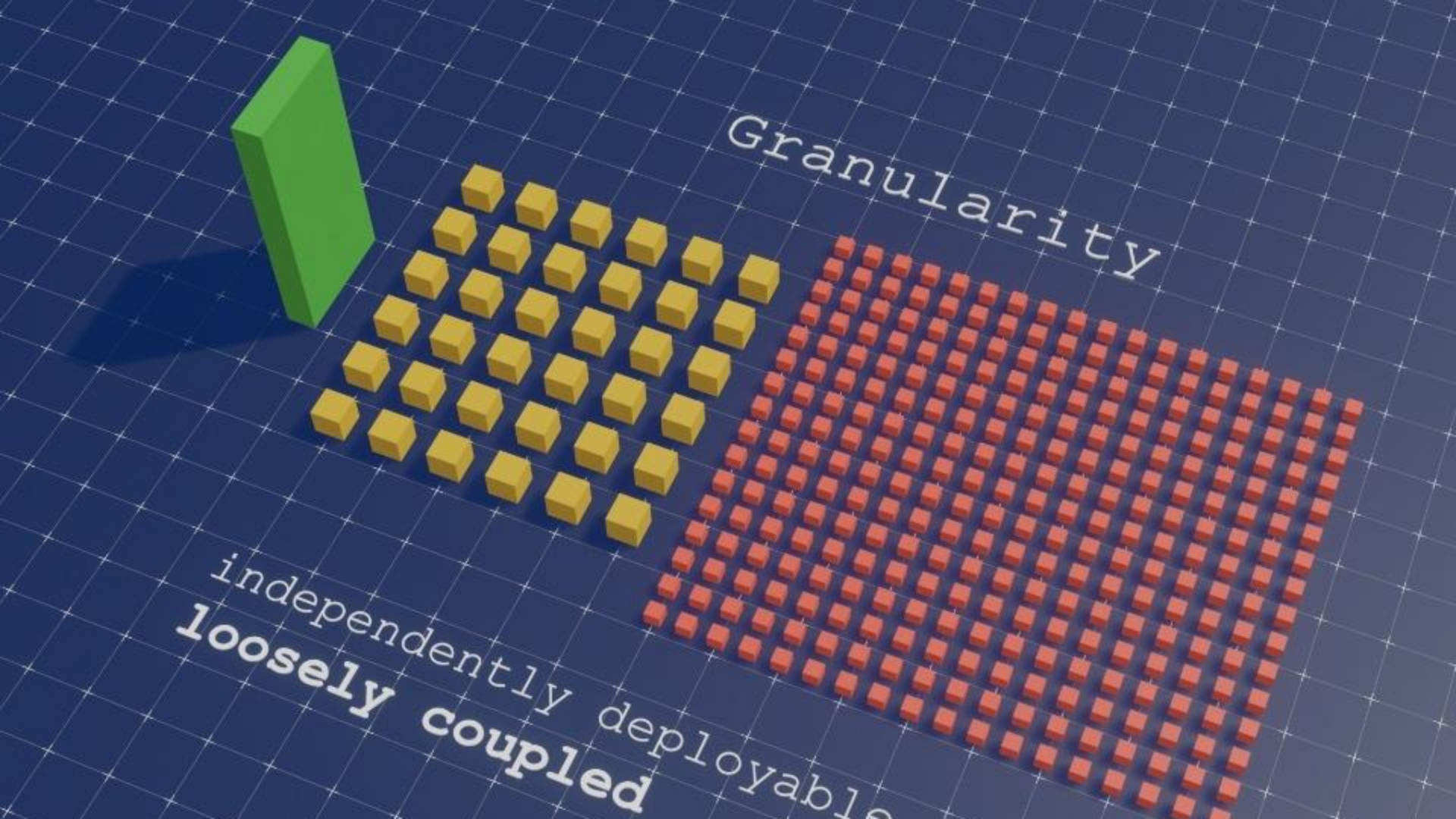


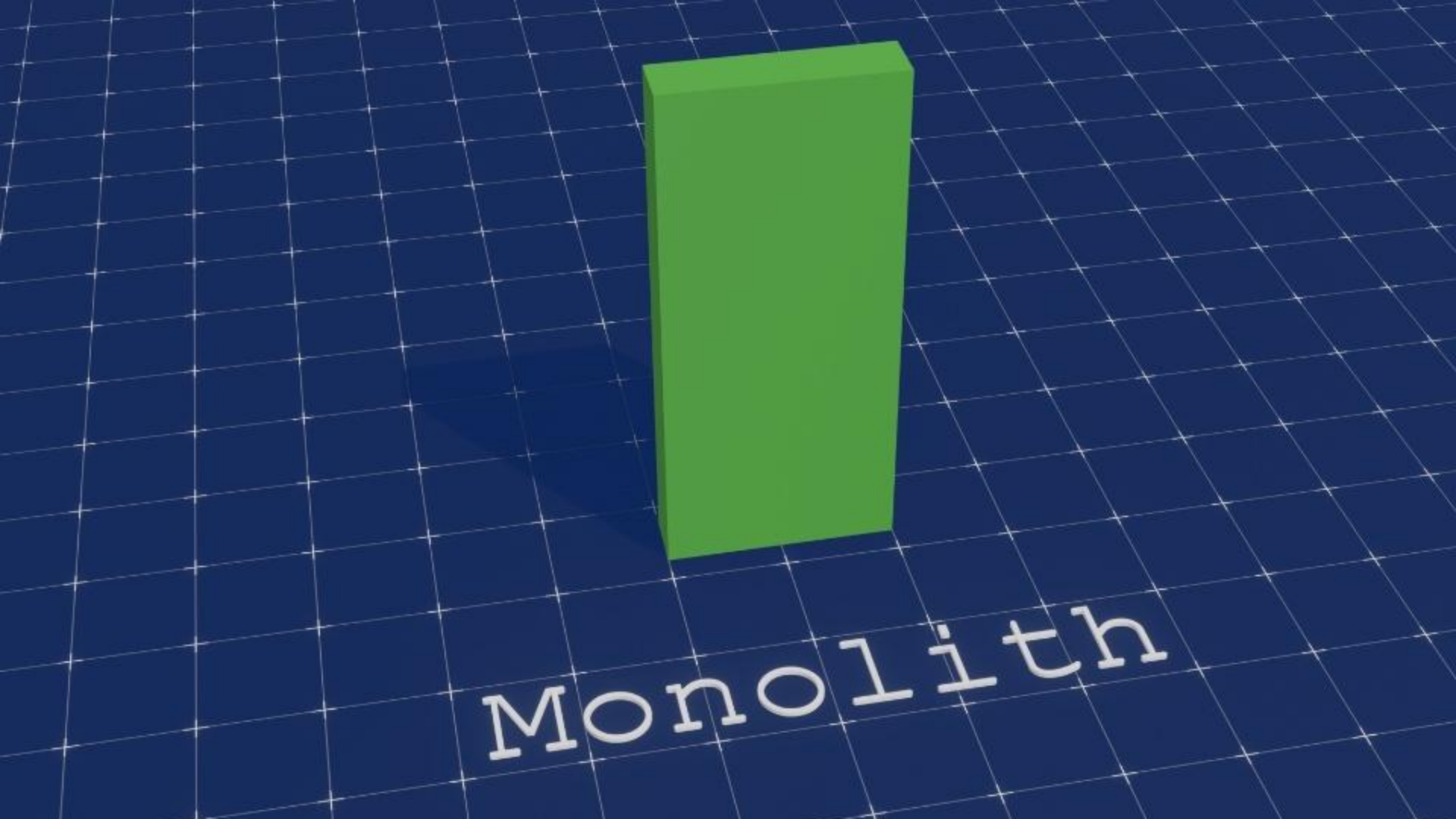

Granularity

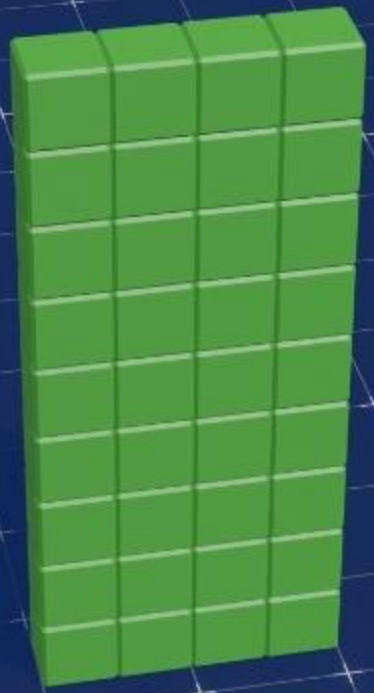

Granularity
Why?


Granularity
Innovation velocity


Granularity
independently deployable

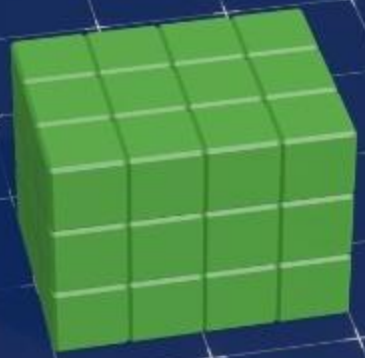

Granularity
independently deployable
loosely coupled


Monolith



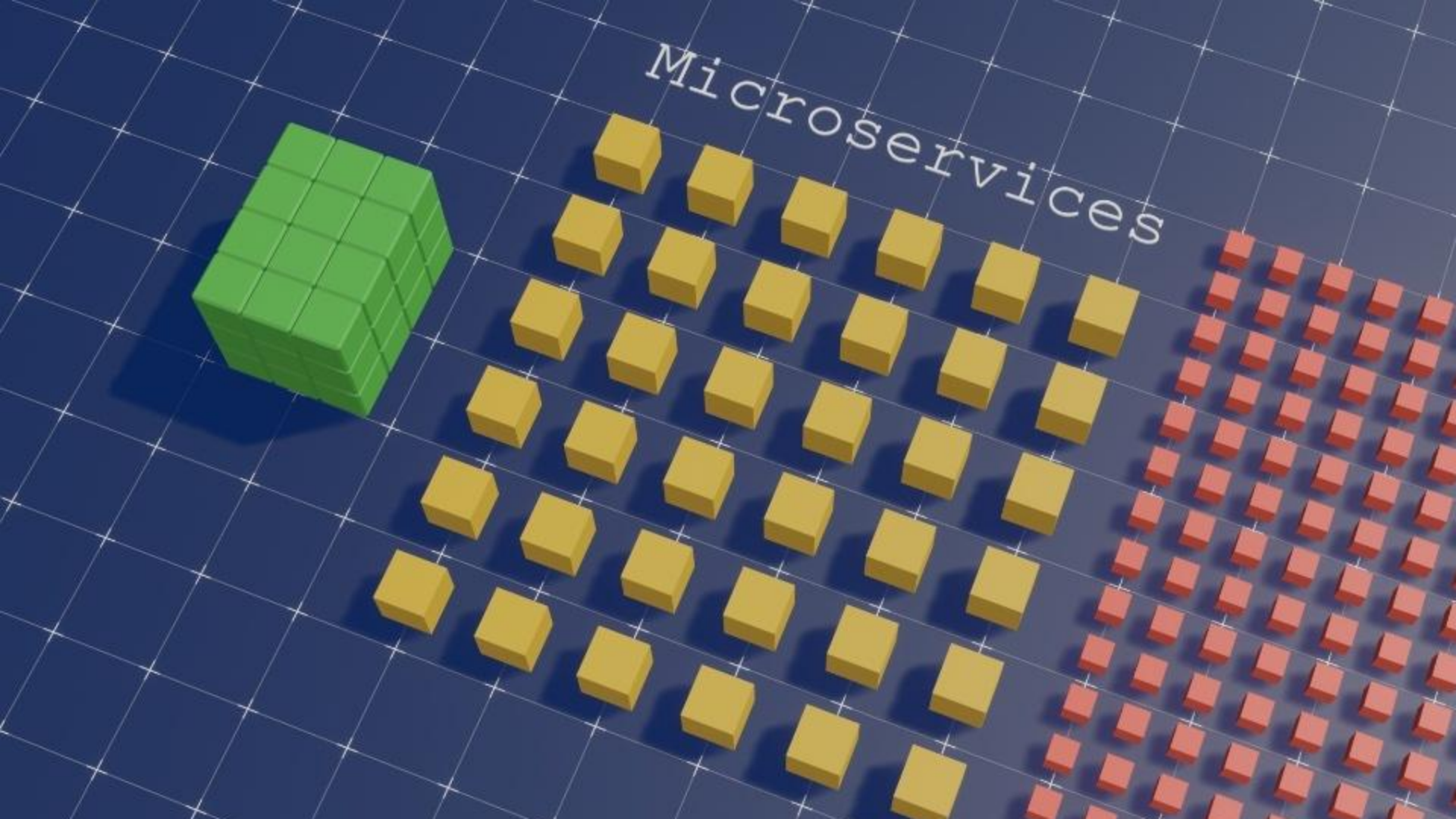
Modules

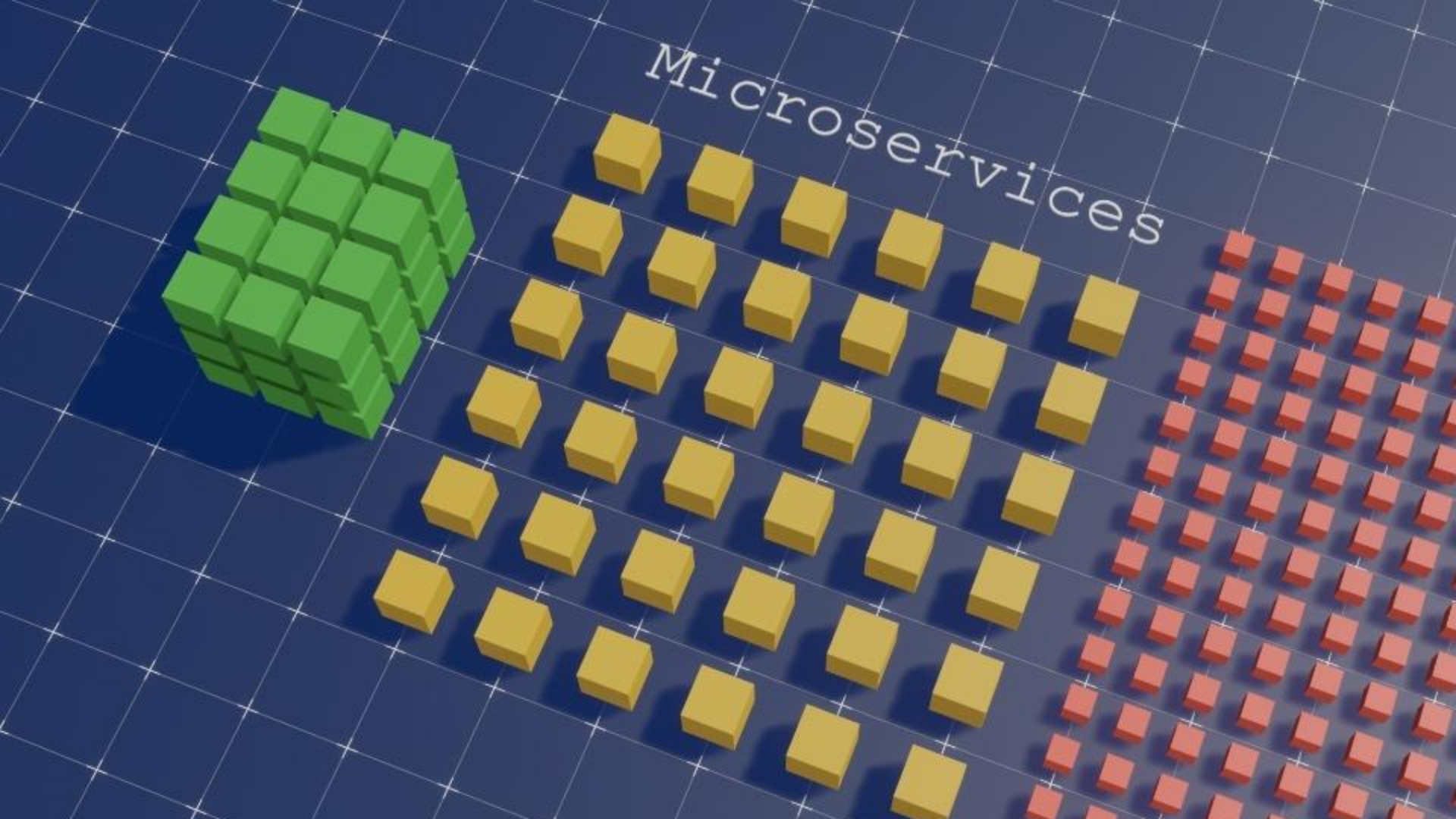
Monolith

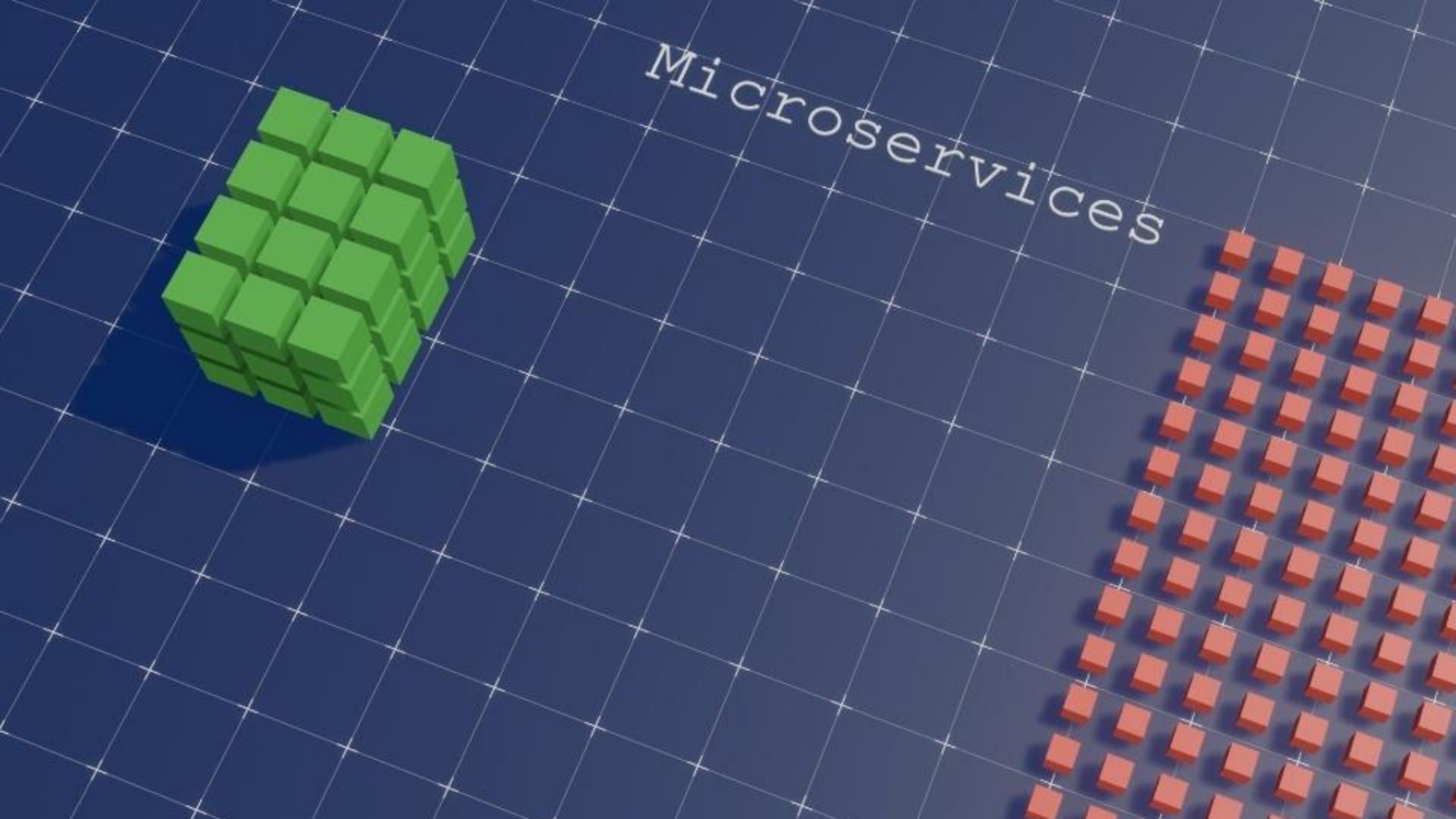


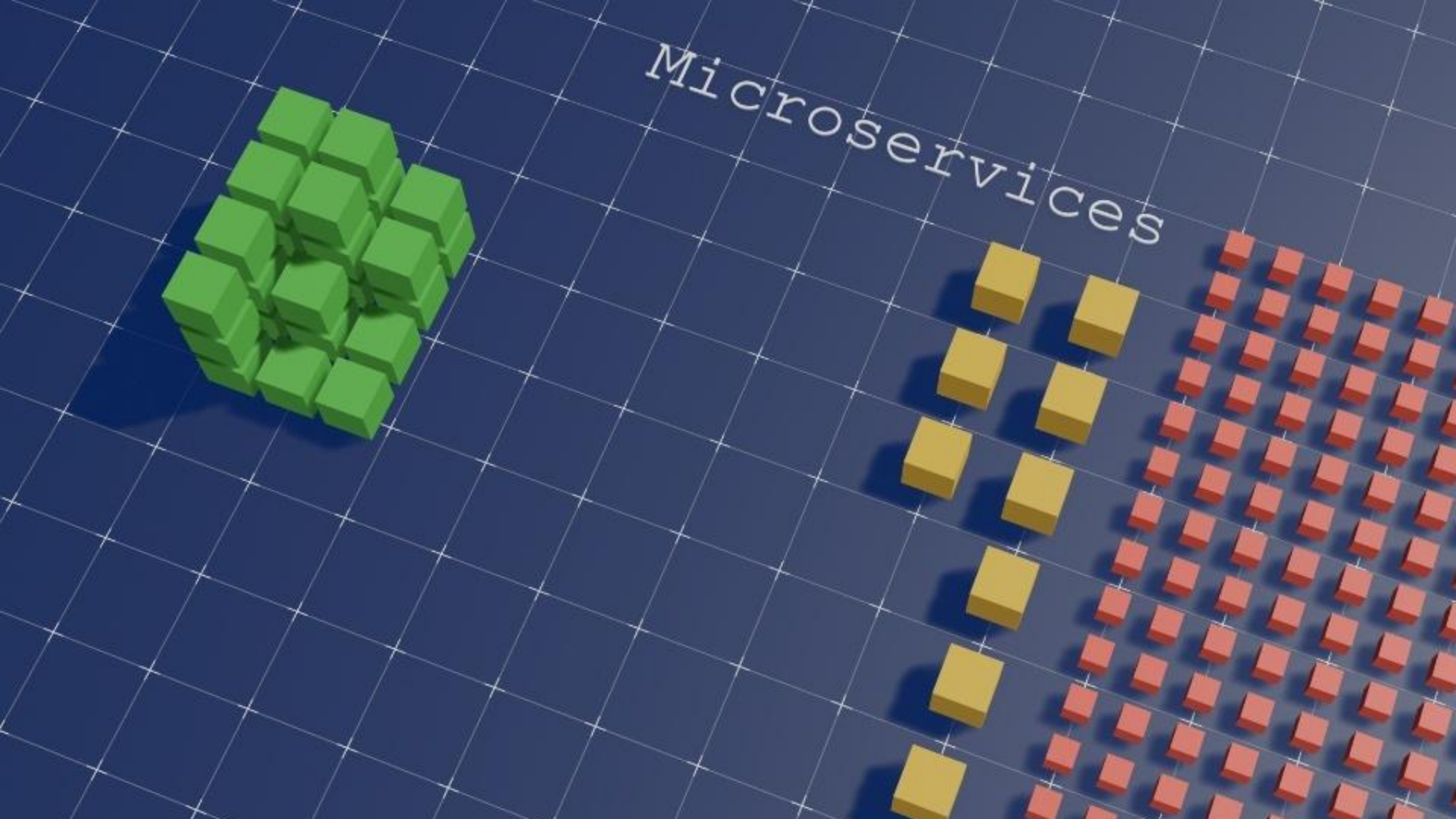
Modules

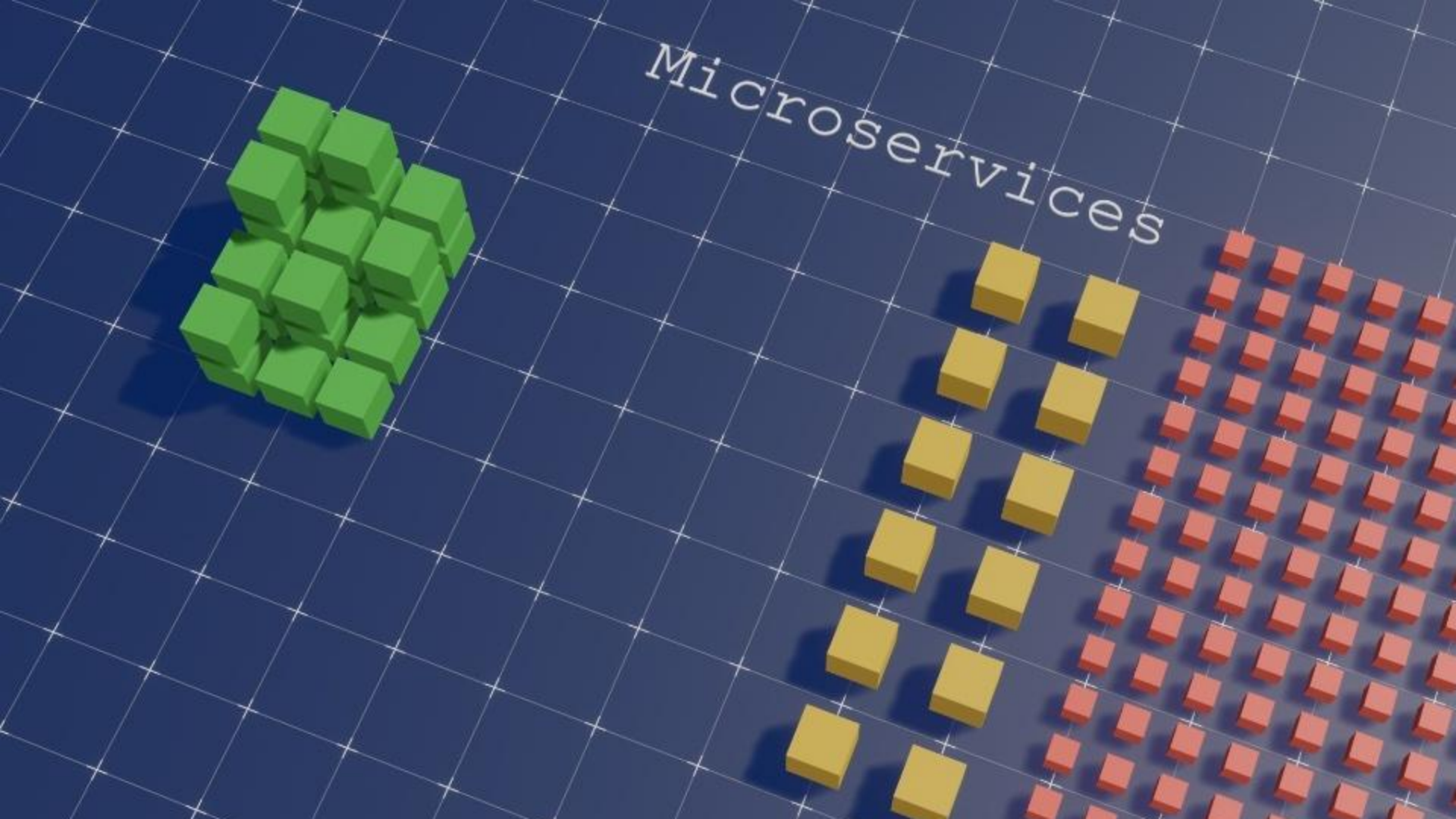
Monolith

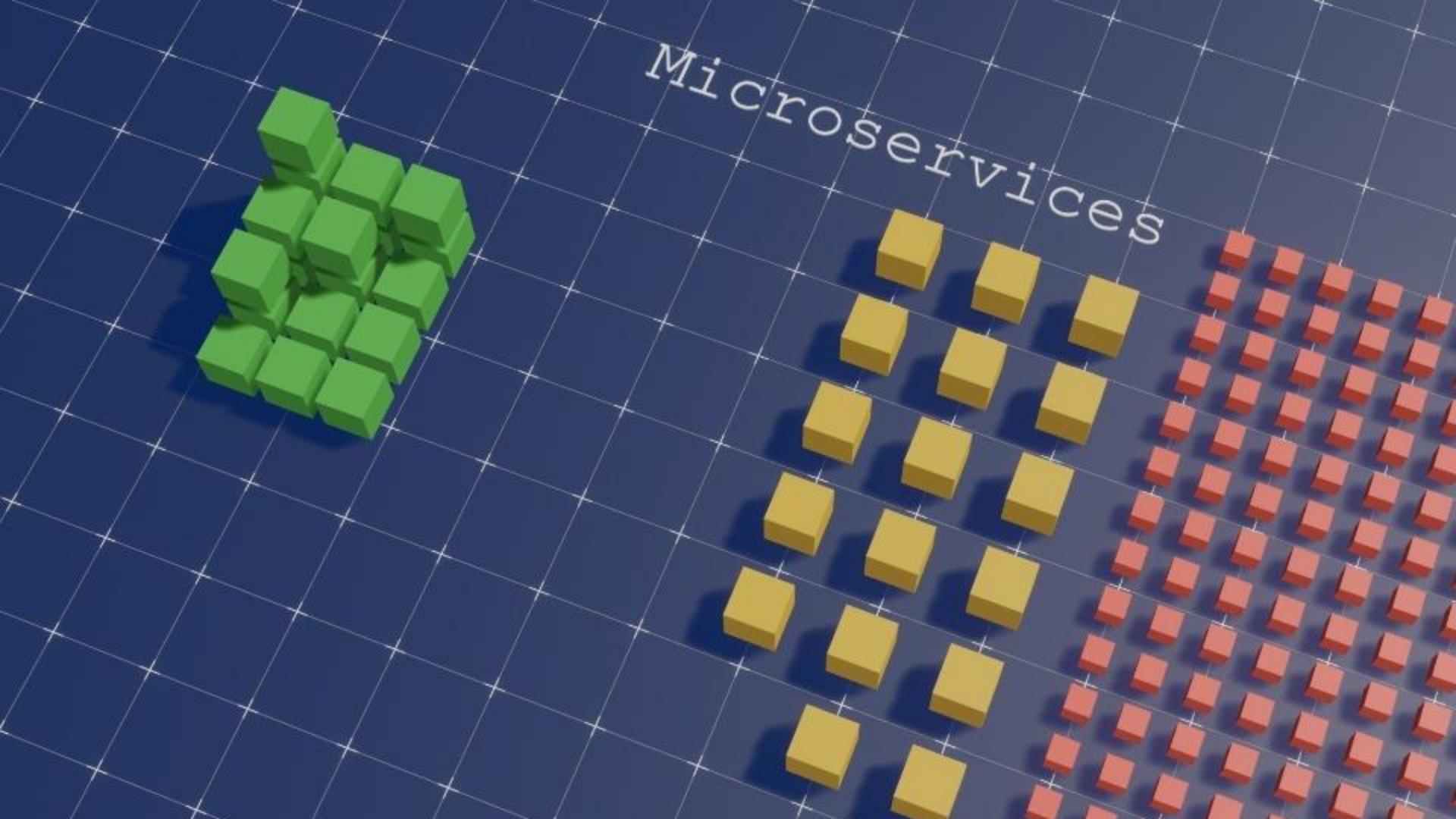

Microservices

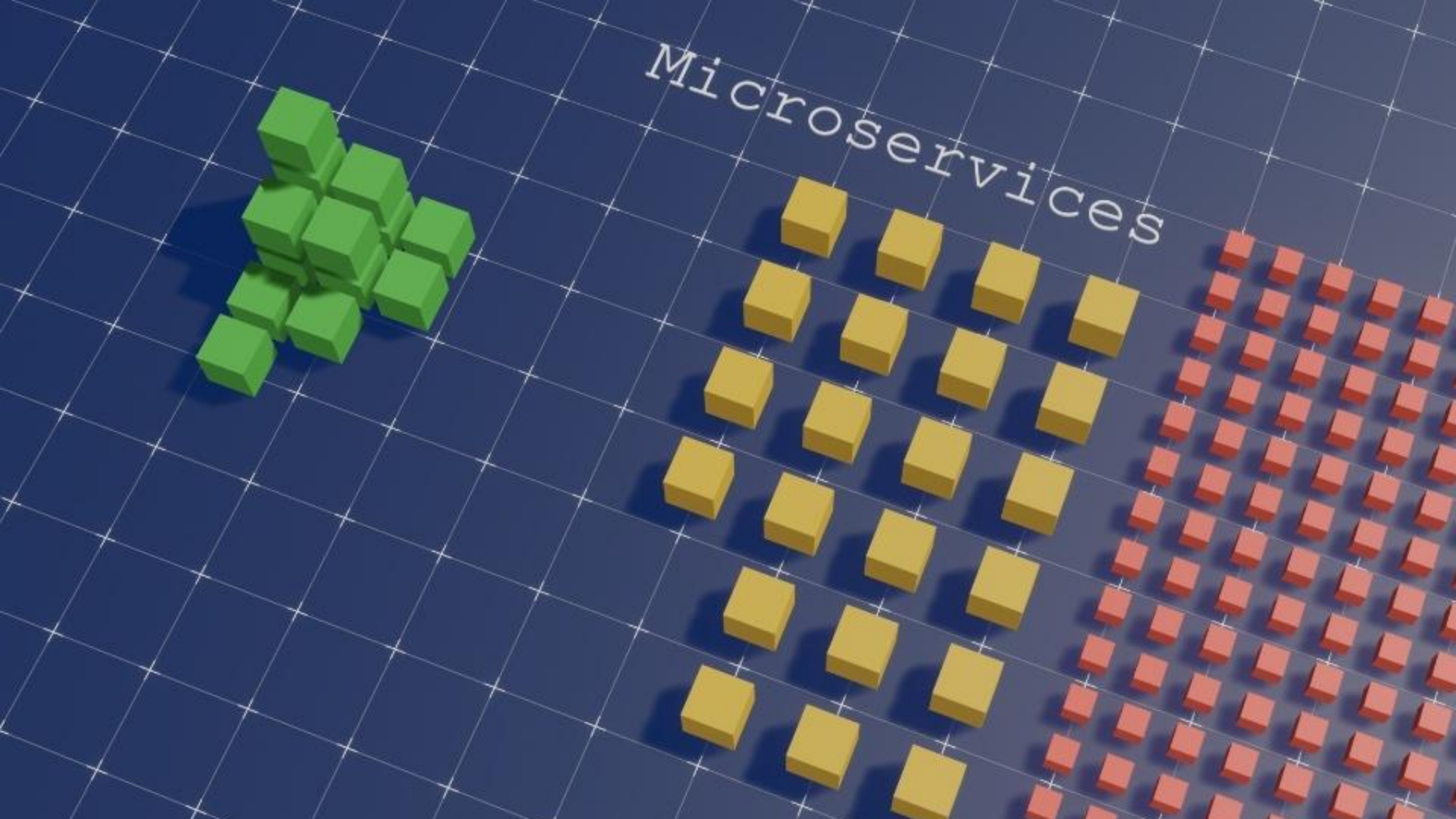

Microservices

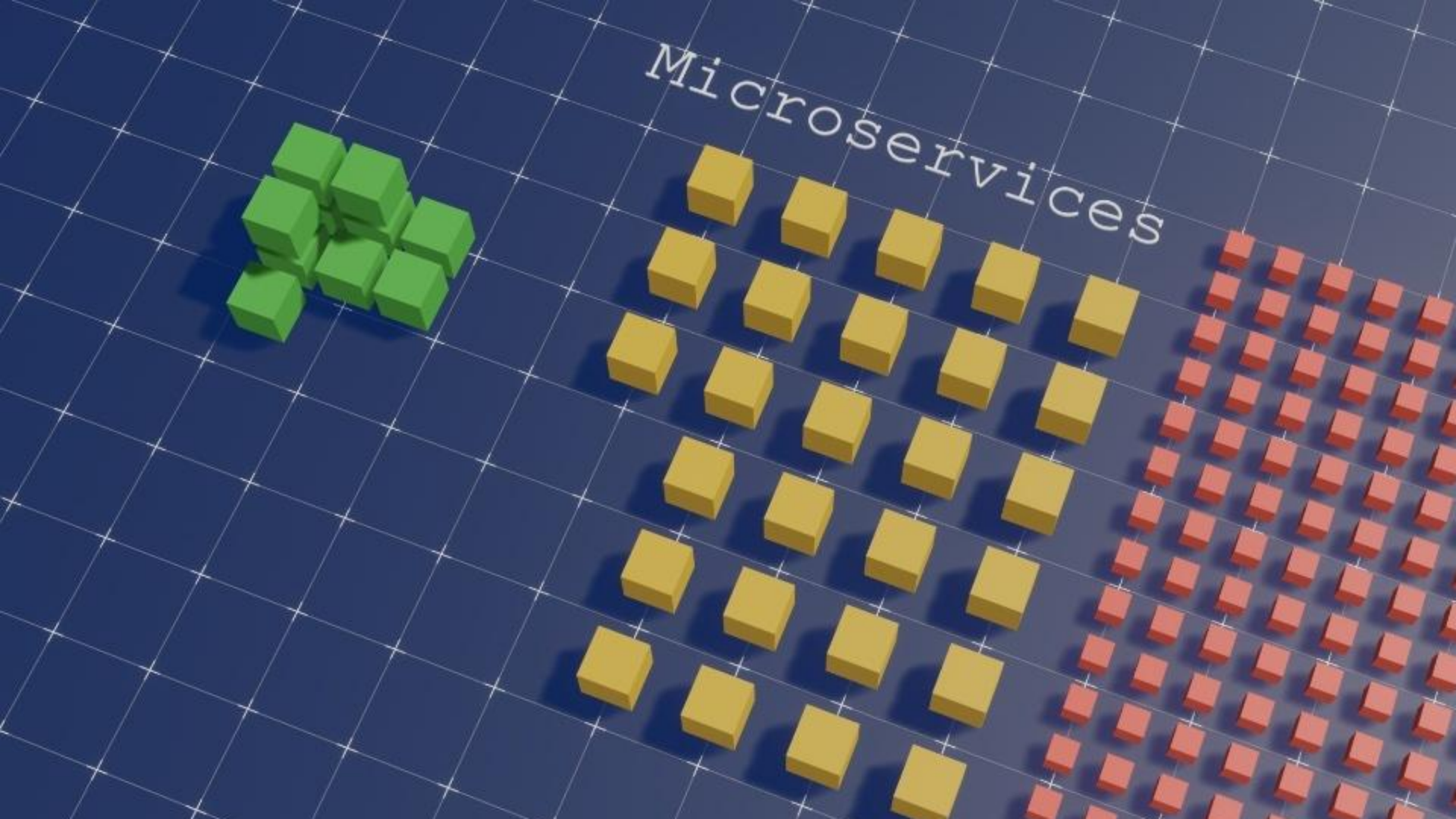

Microservices

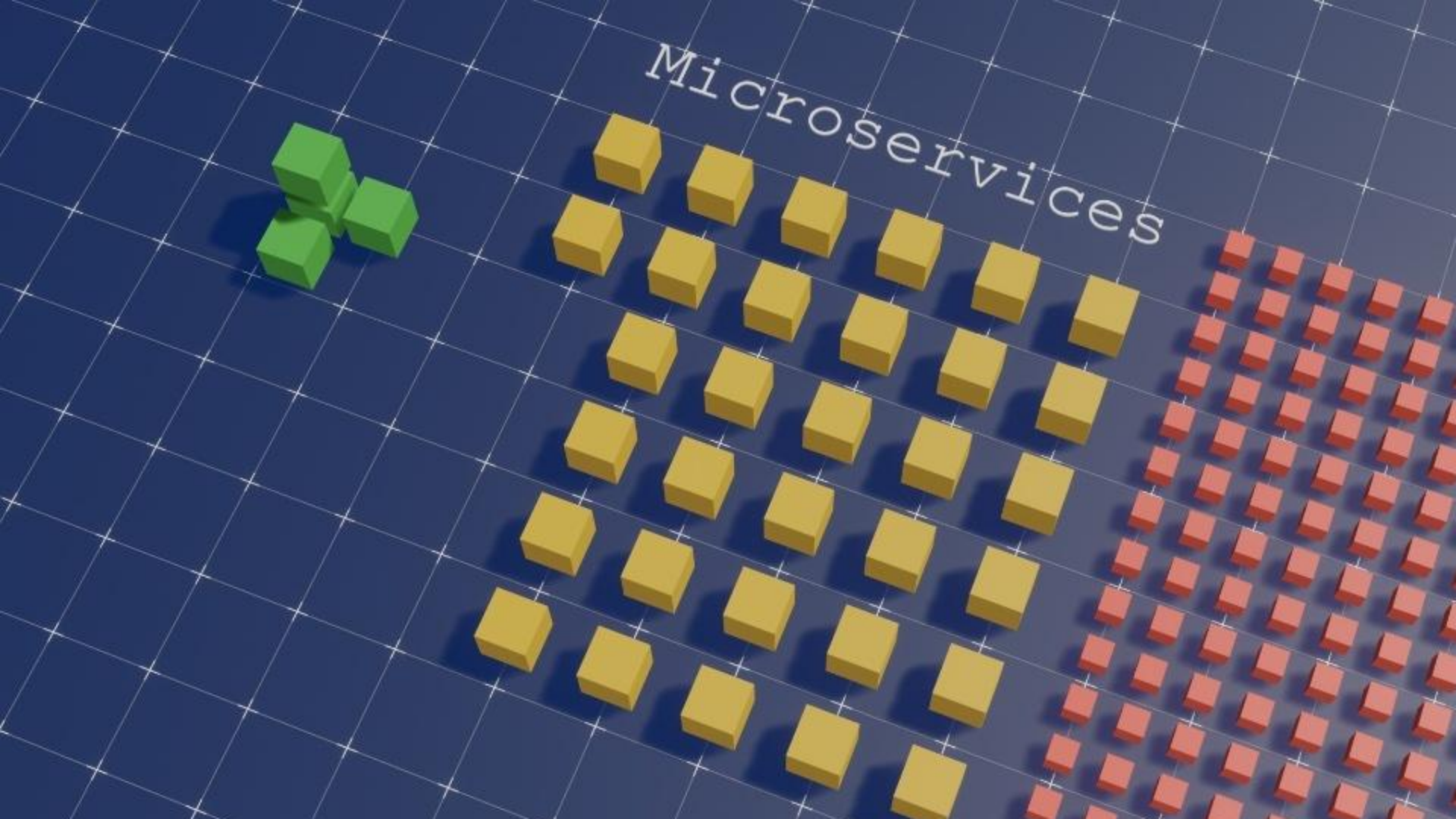

Microservices

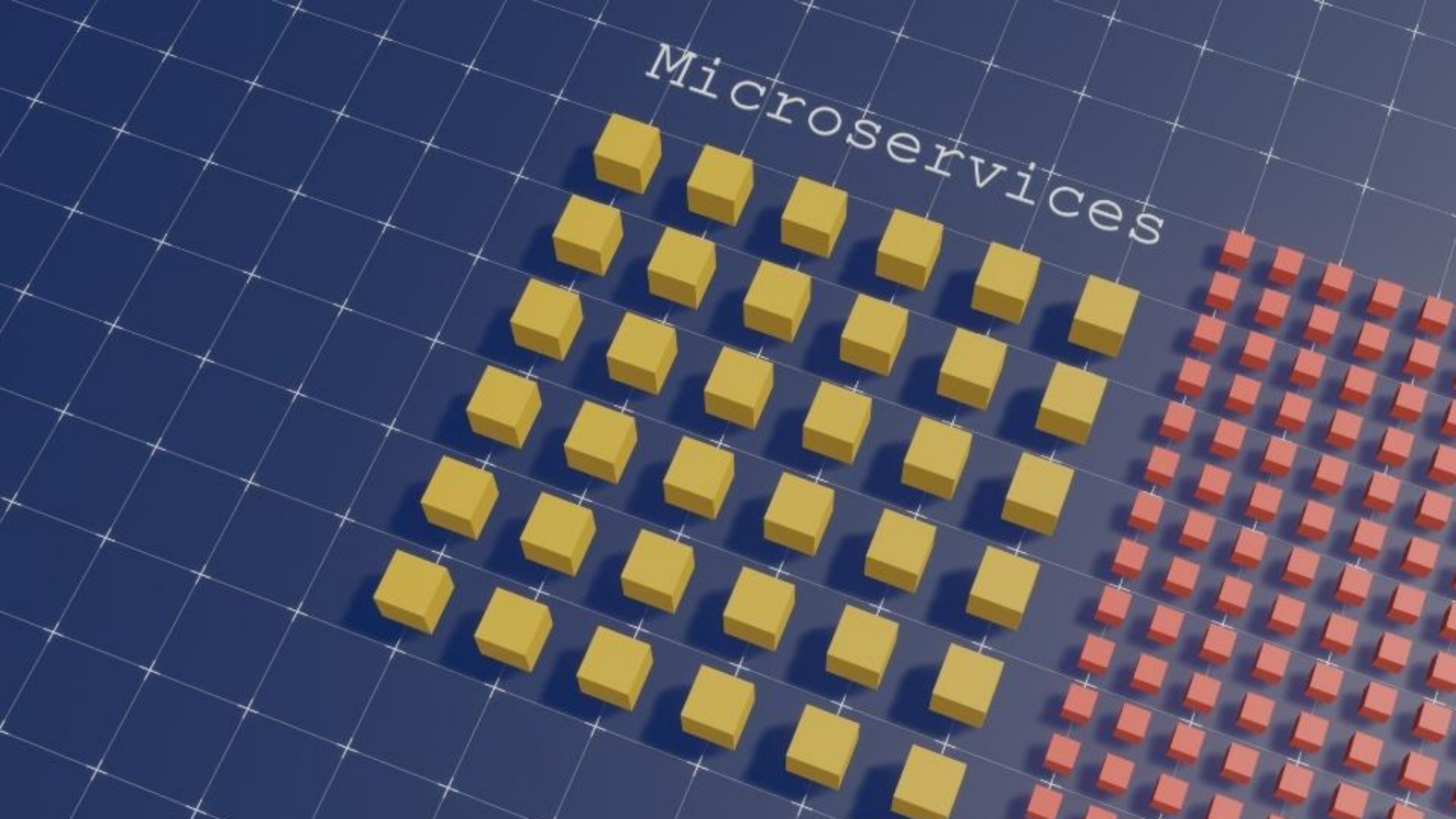

Microservices

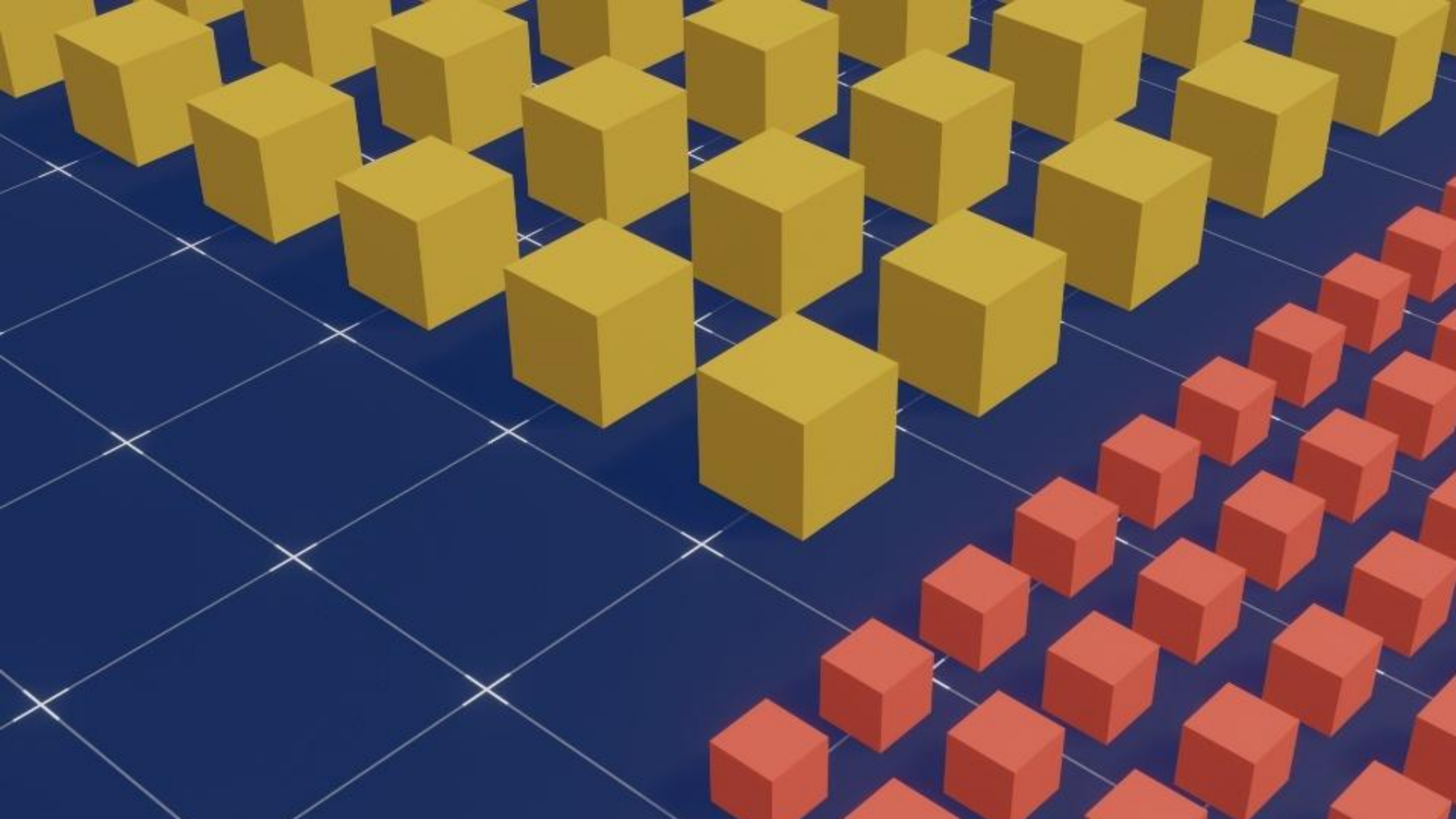

Microservices

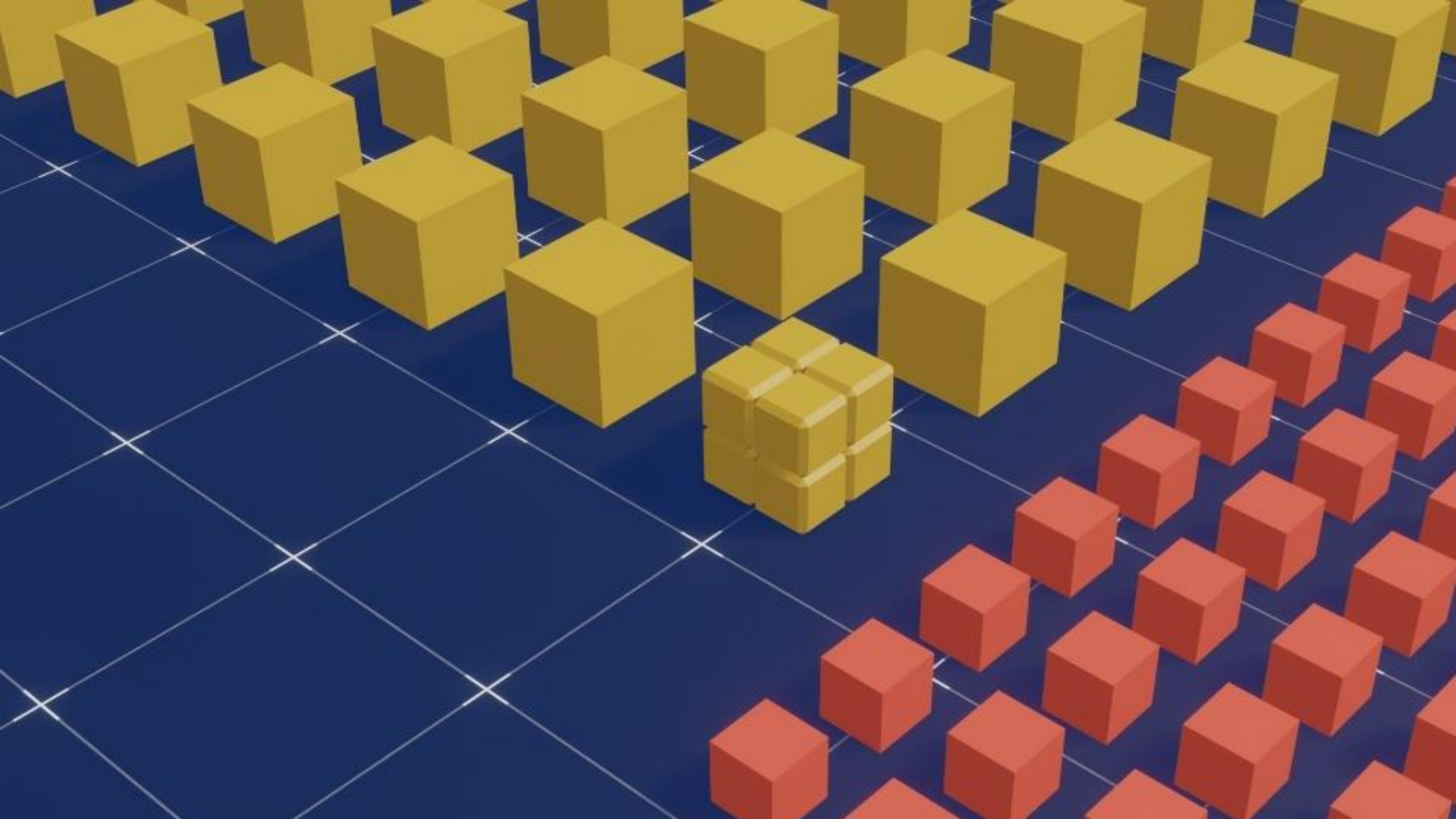

Microservices

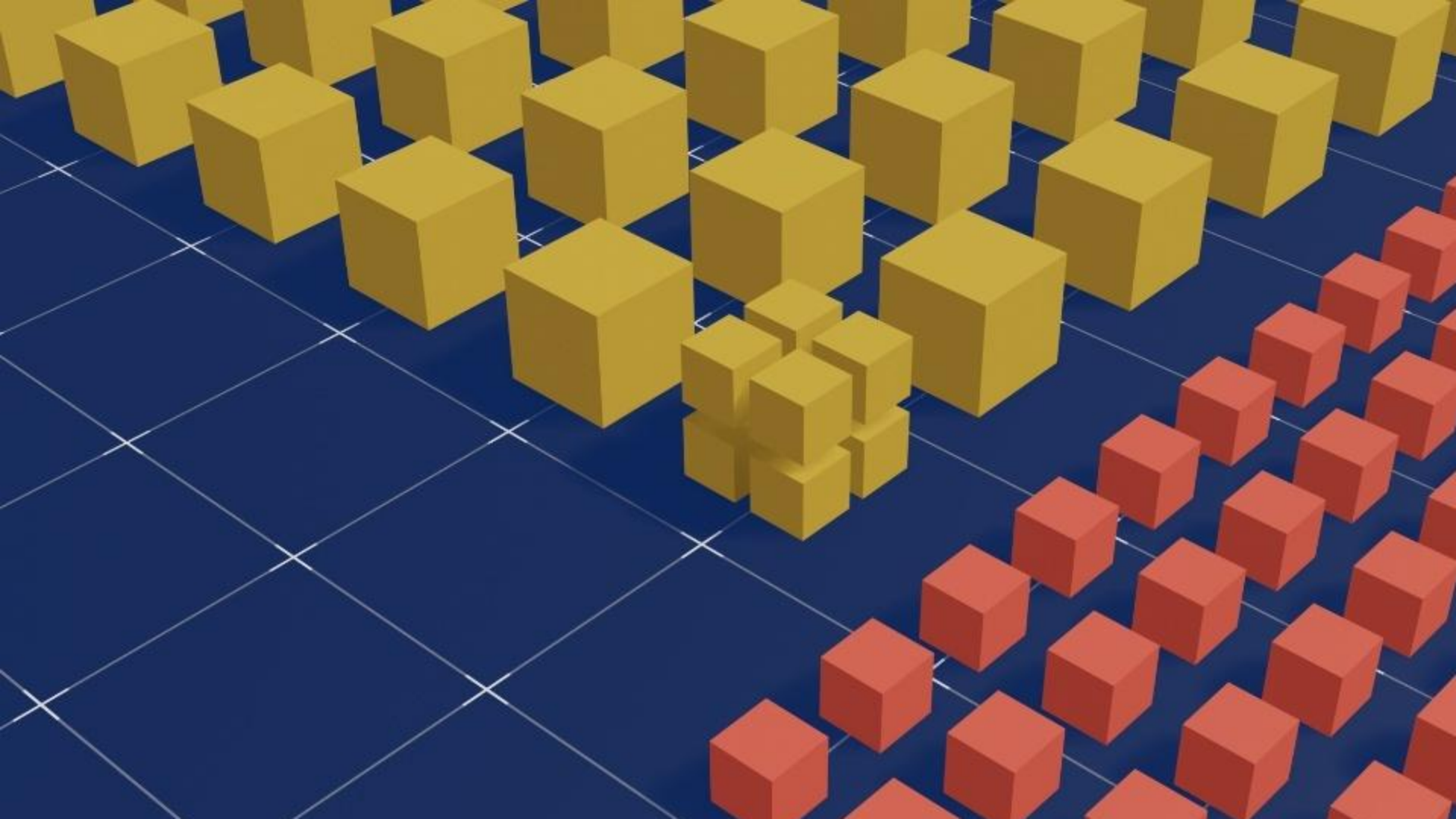

Microservices

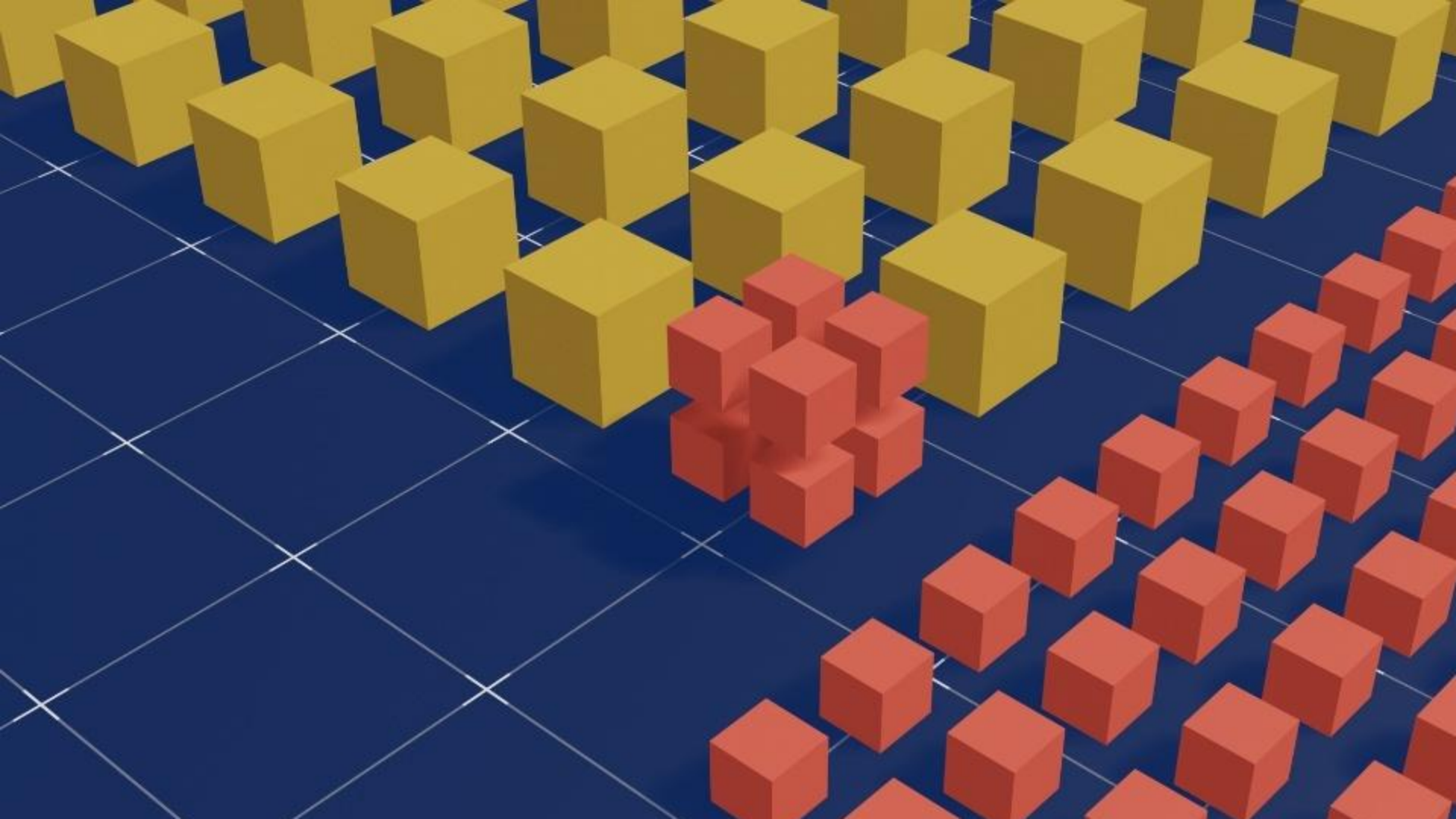

Microservices

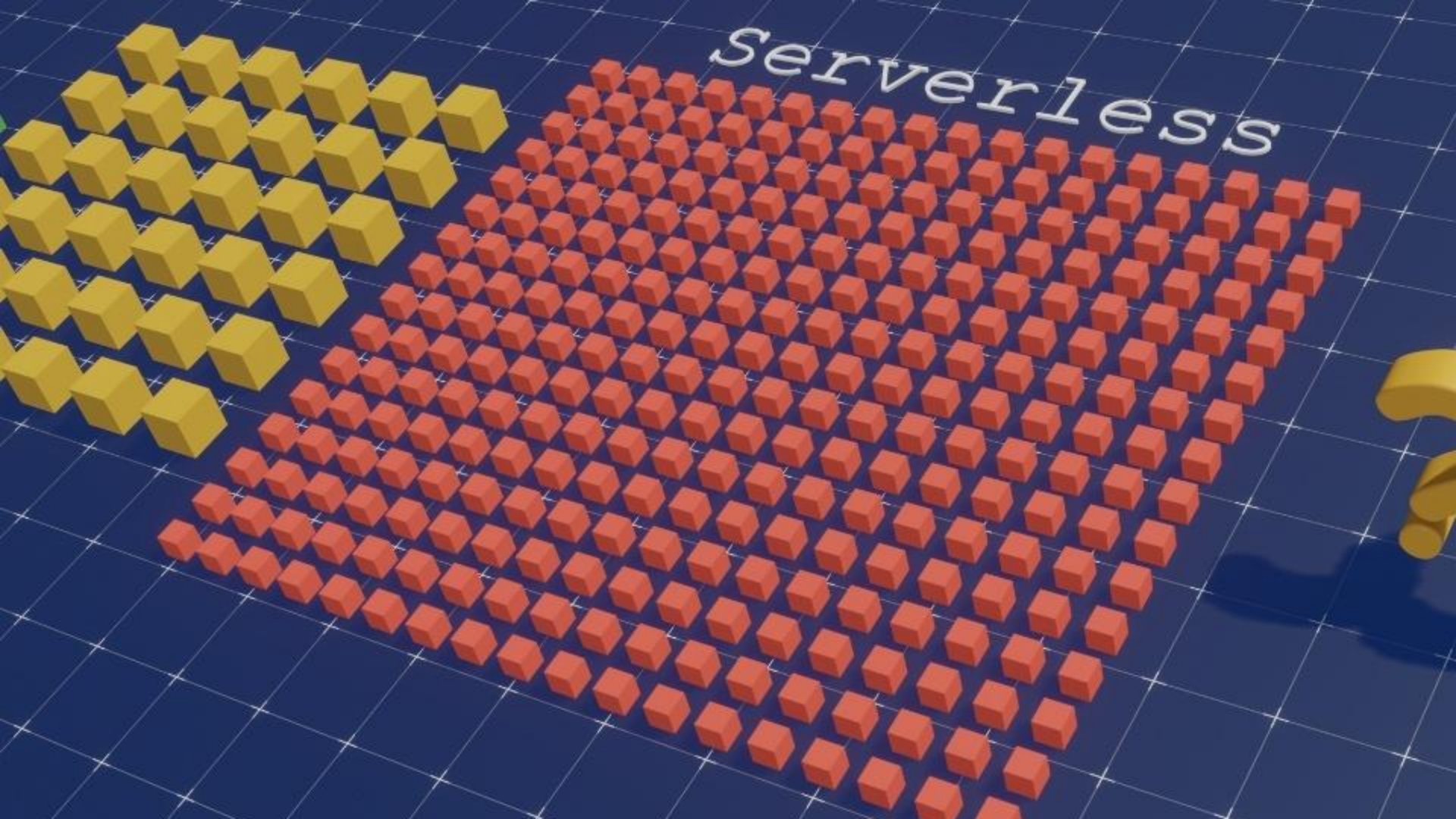

Microservices

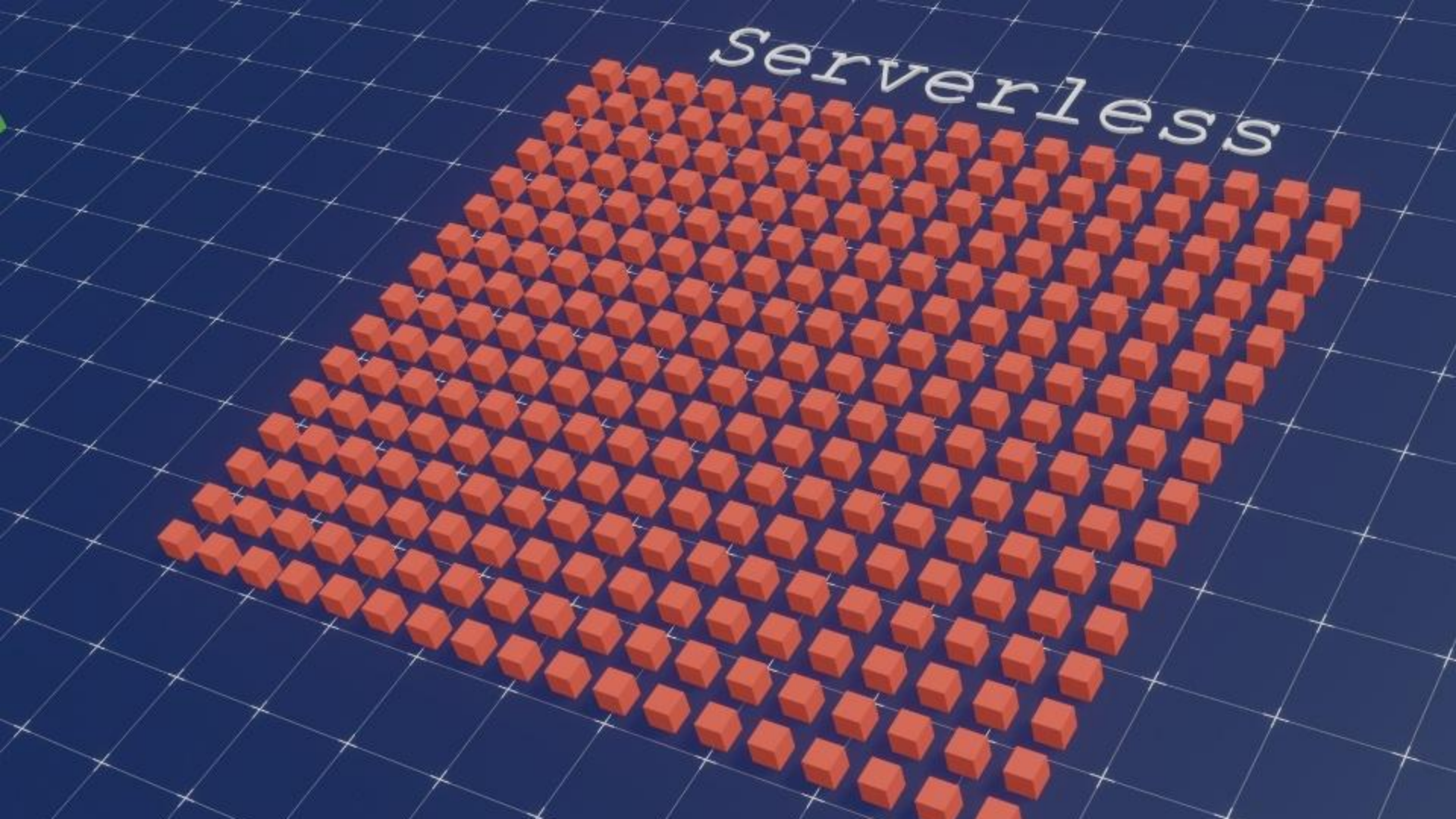


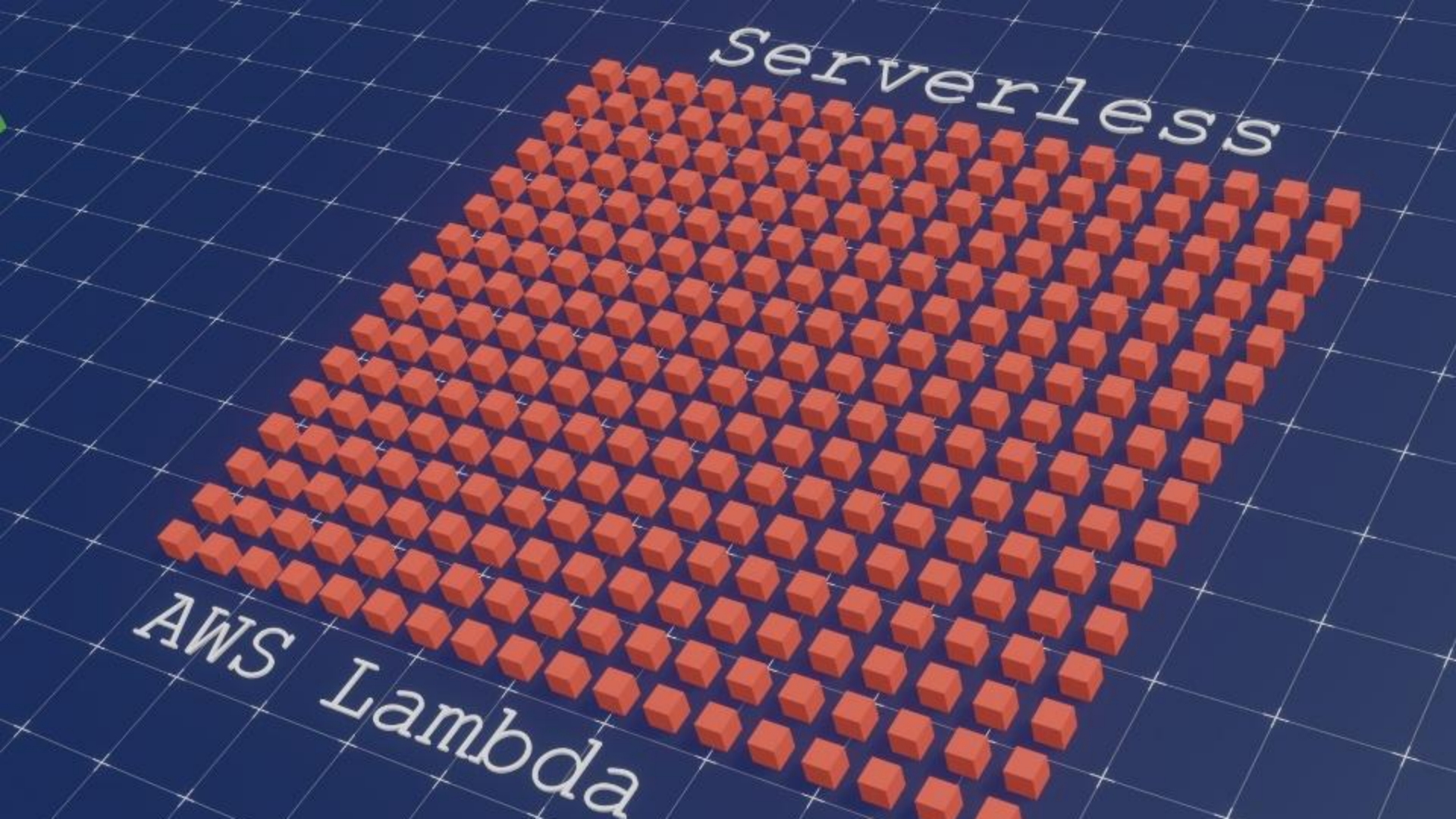


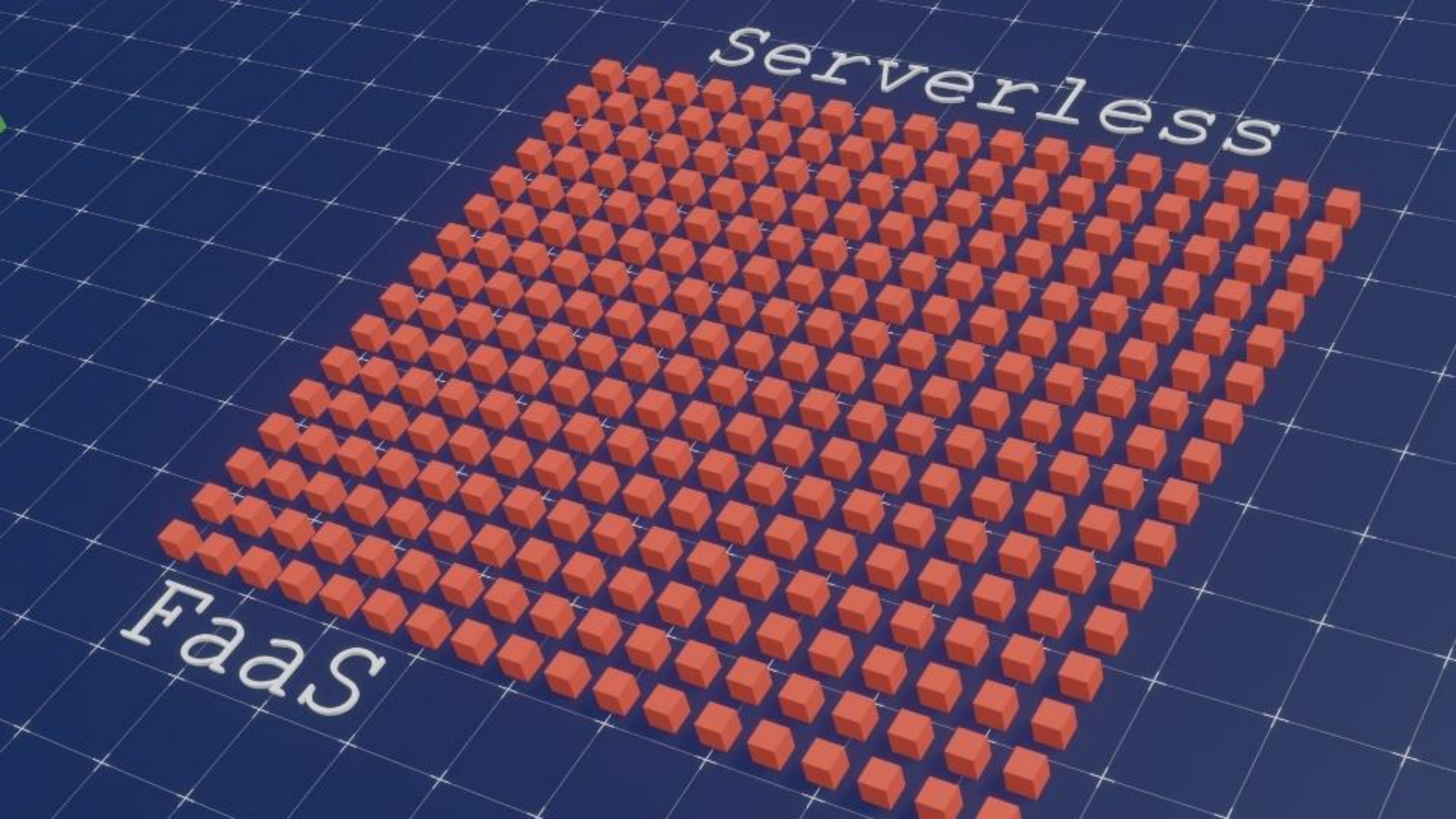


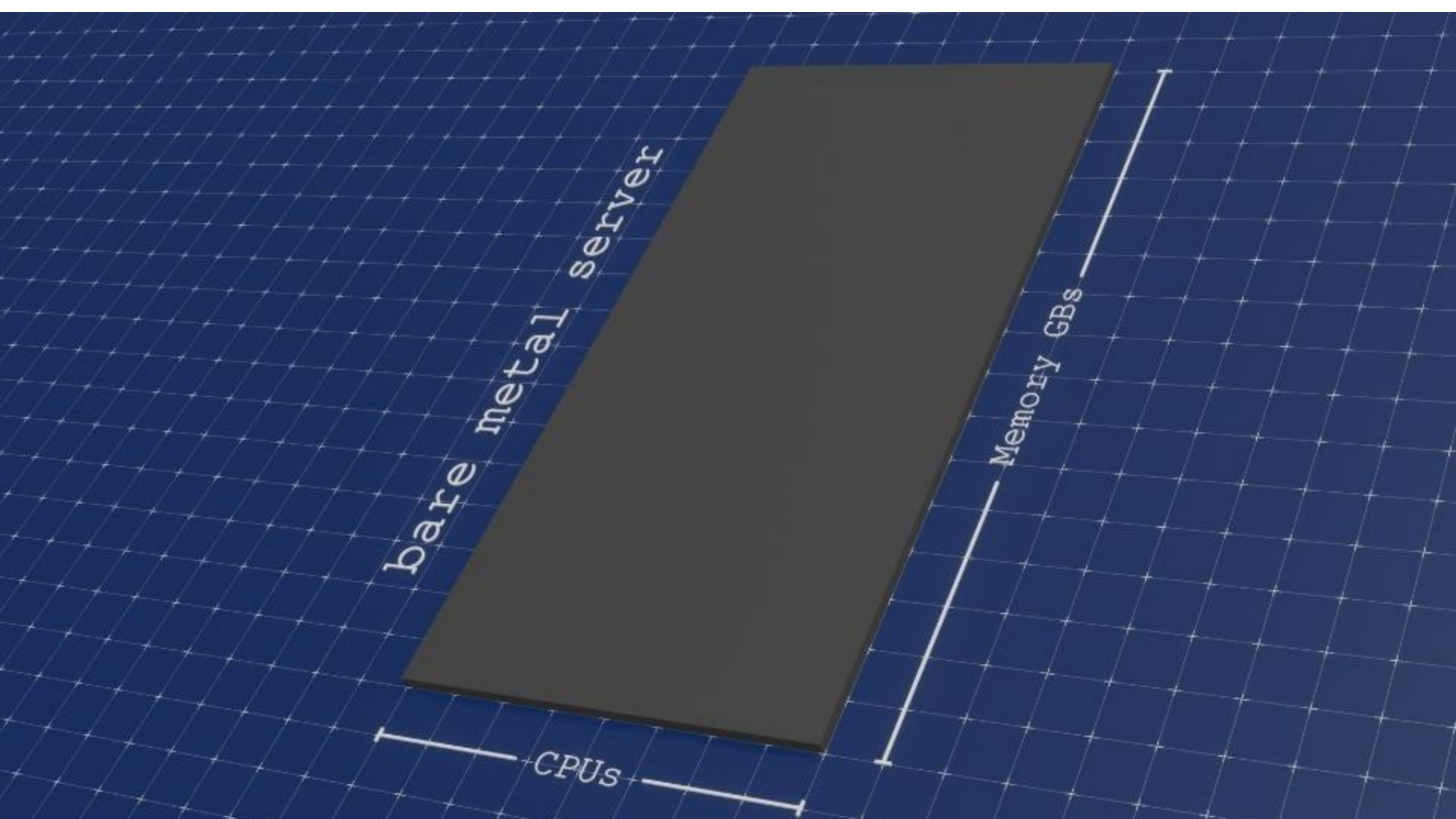


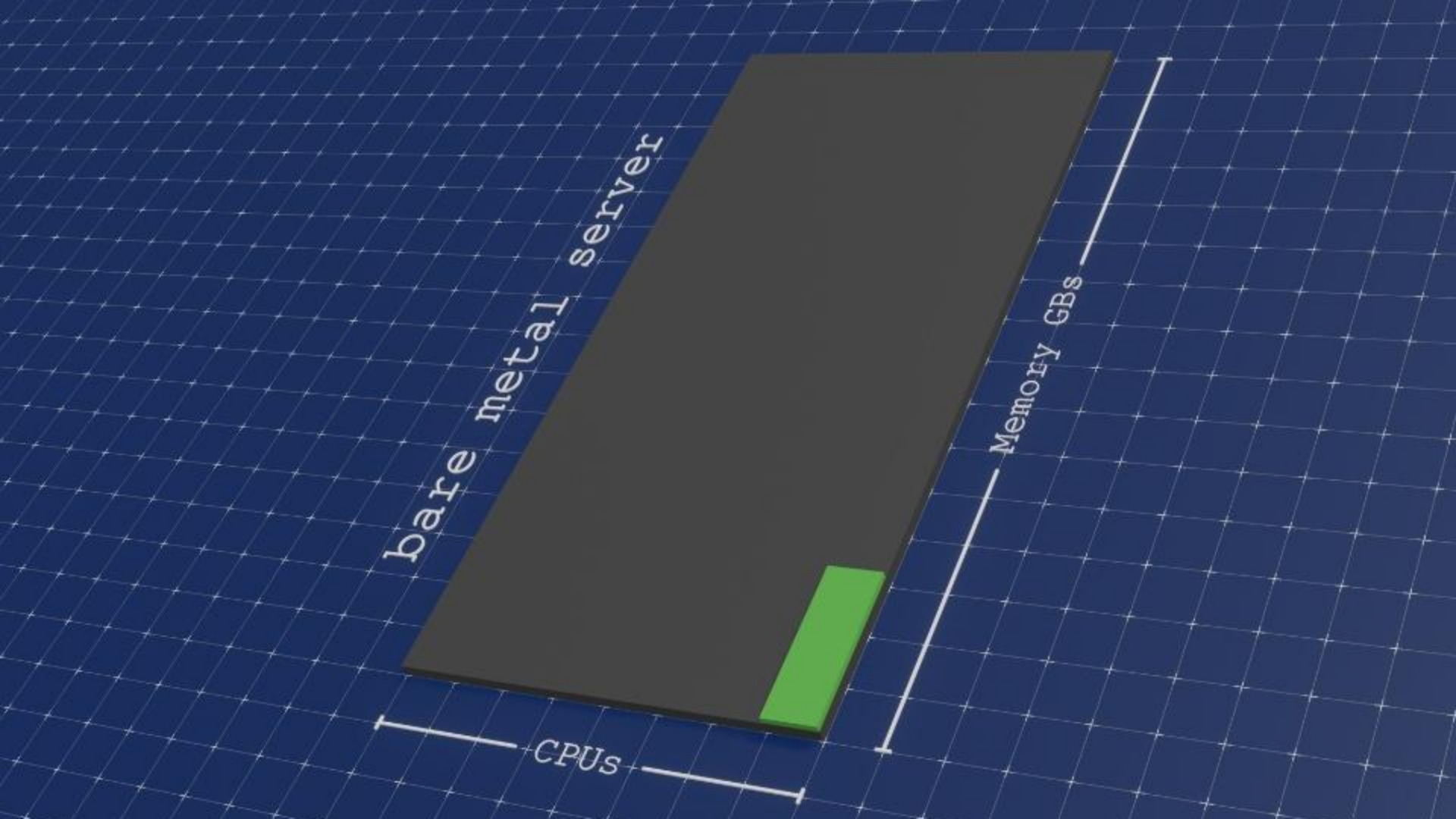

Serverless

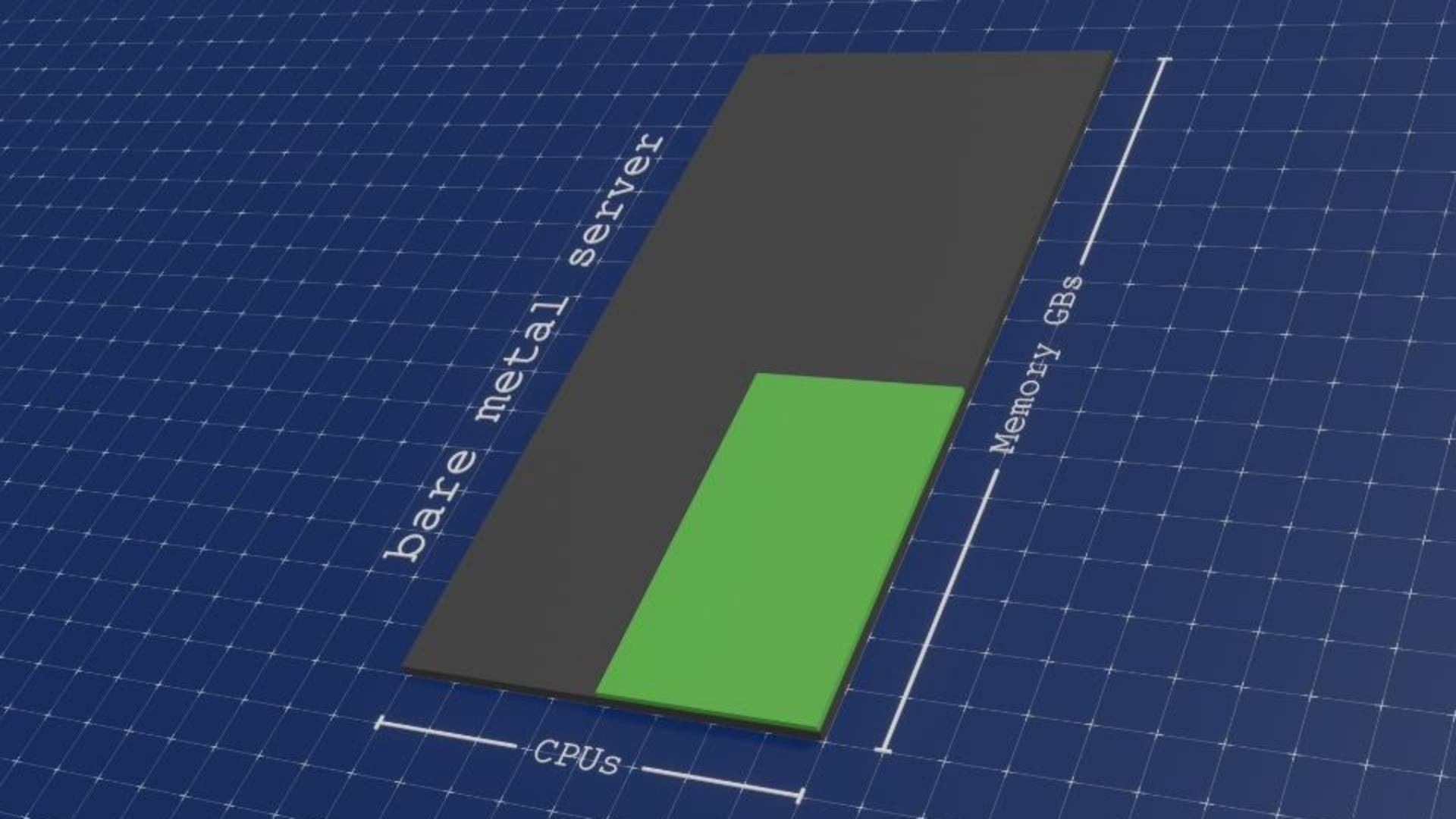

Serverless

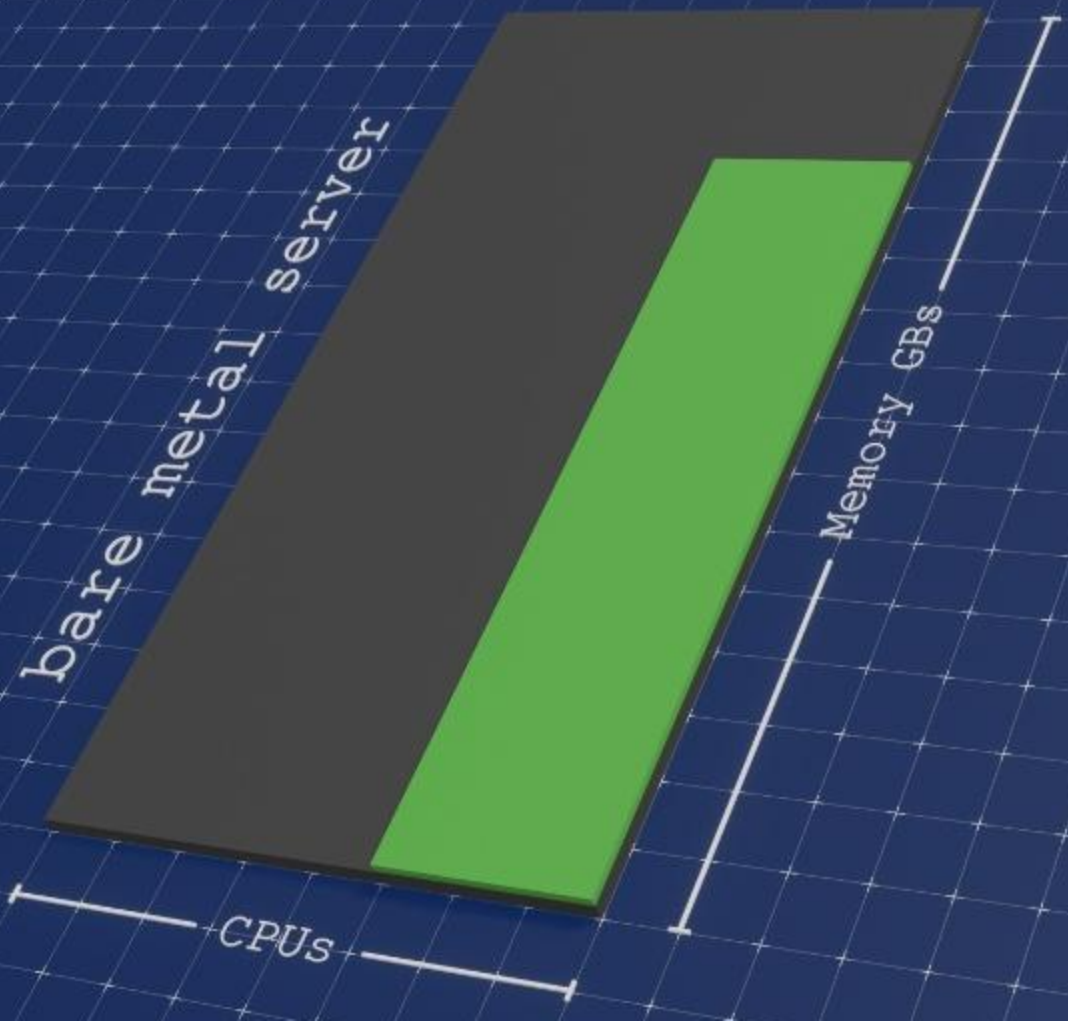

Serverless
AWS Lambda

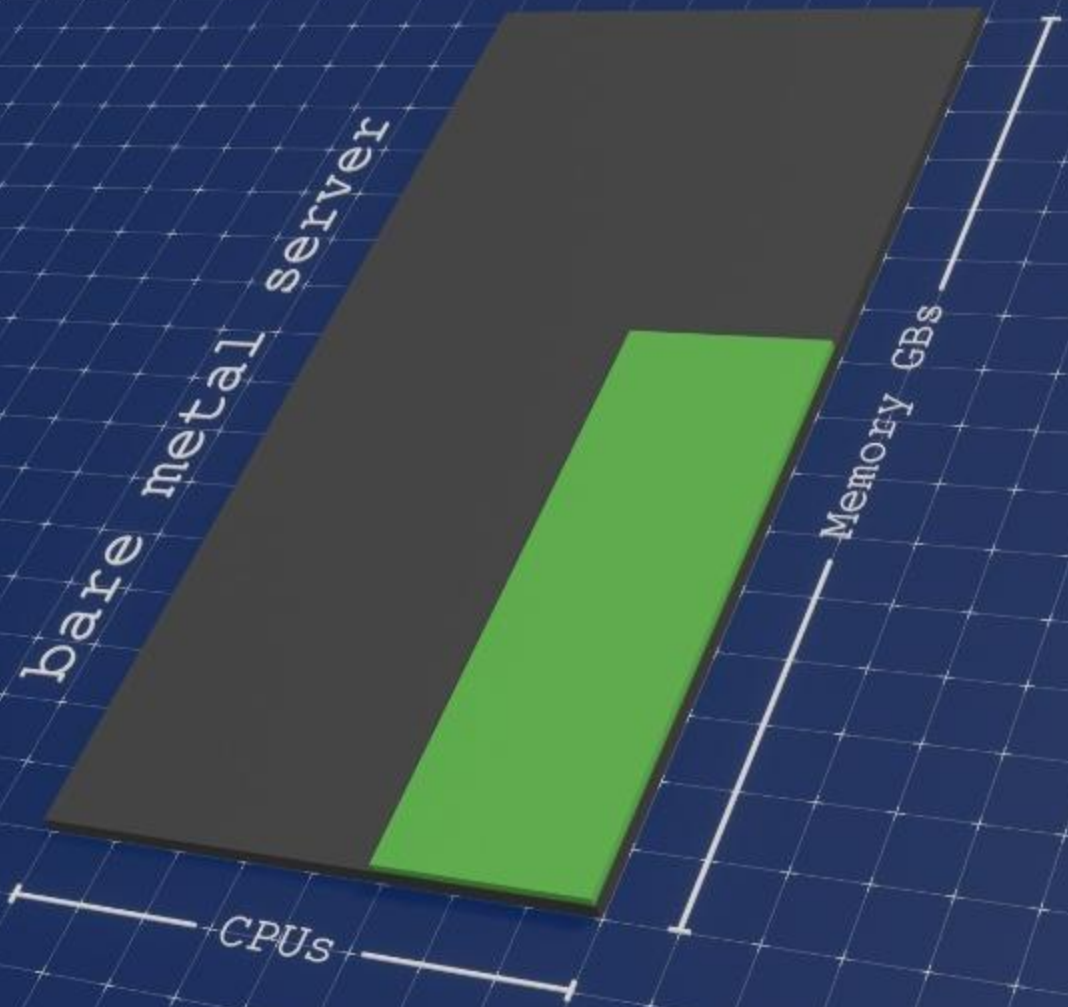

Serverless
FaaS

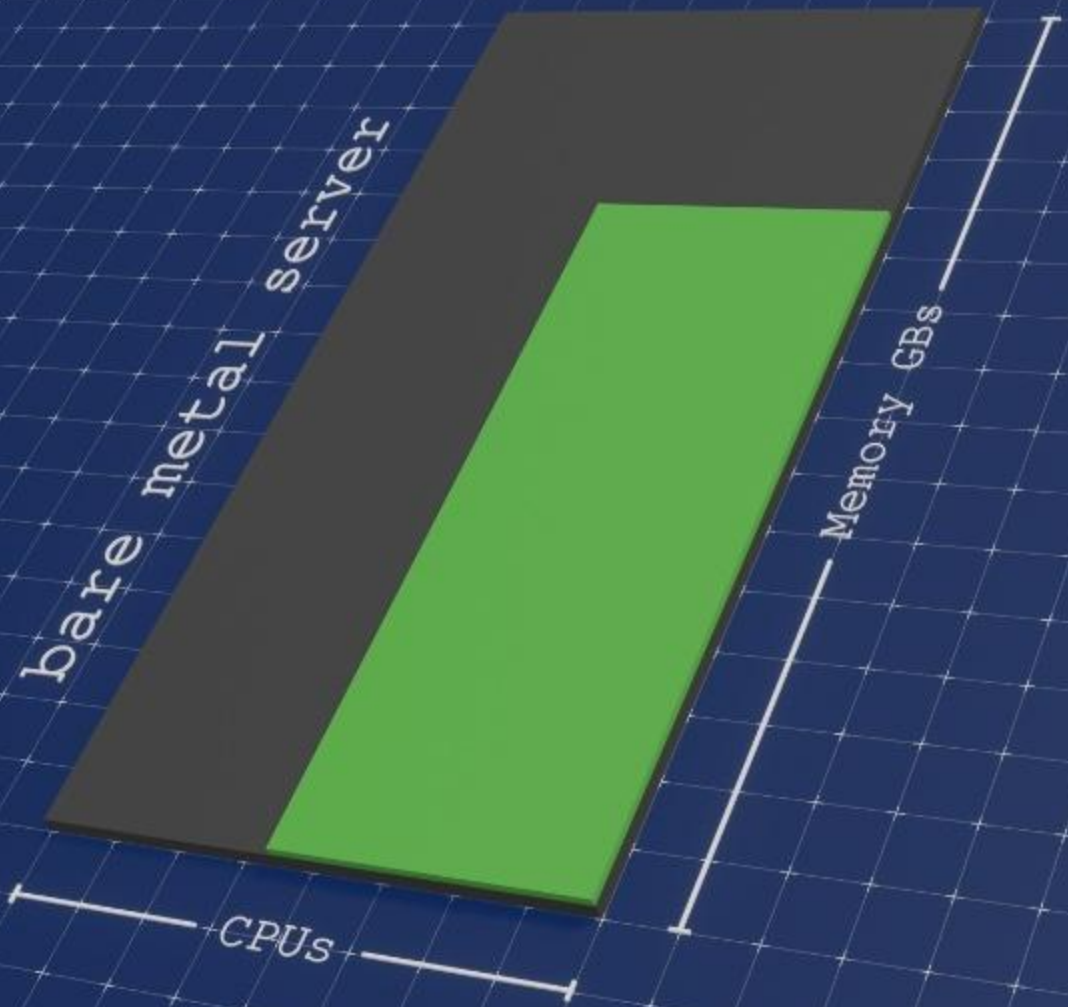

bare metal server
Memory GBs
CPUs

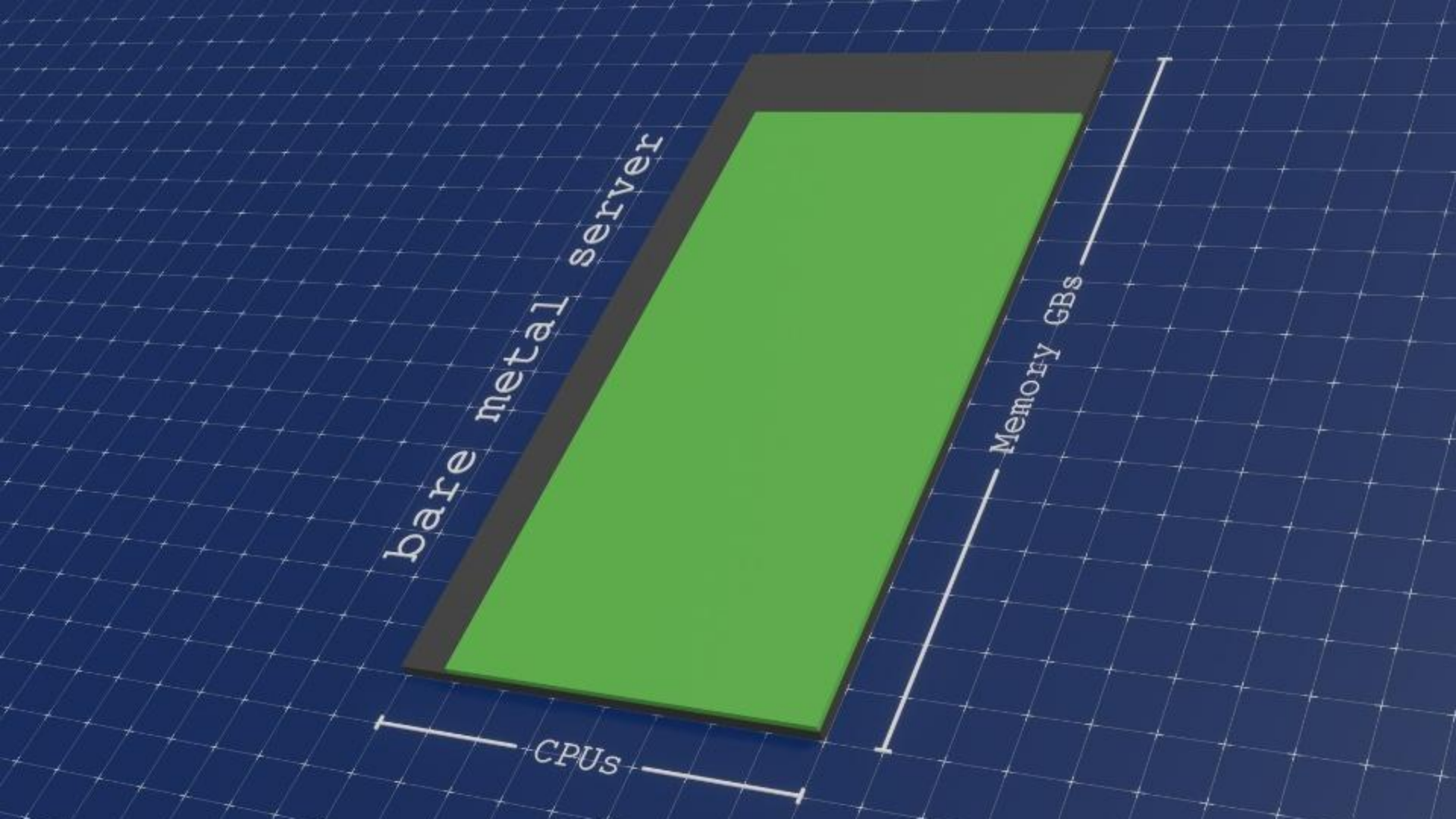

bare metal server
Memory GBs
CPUs

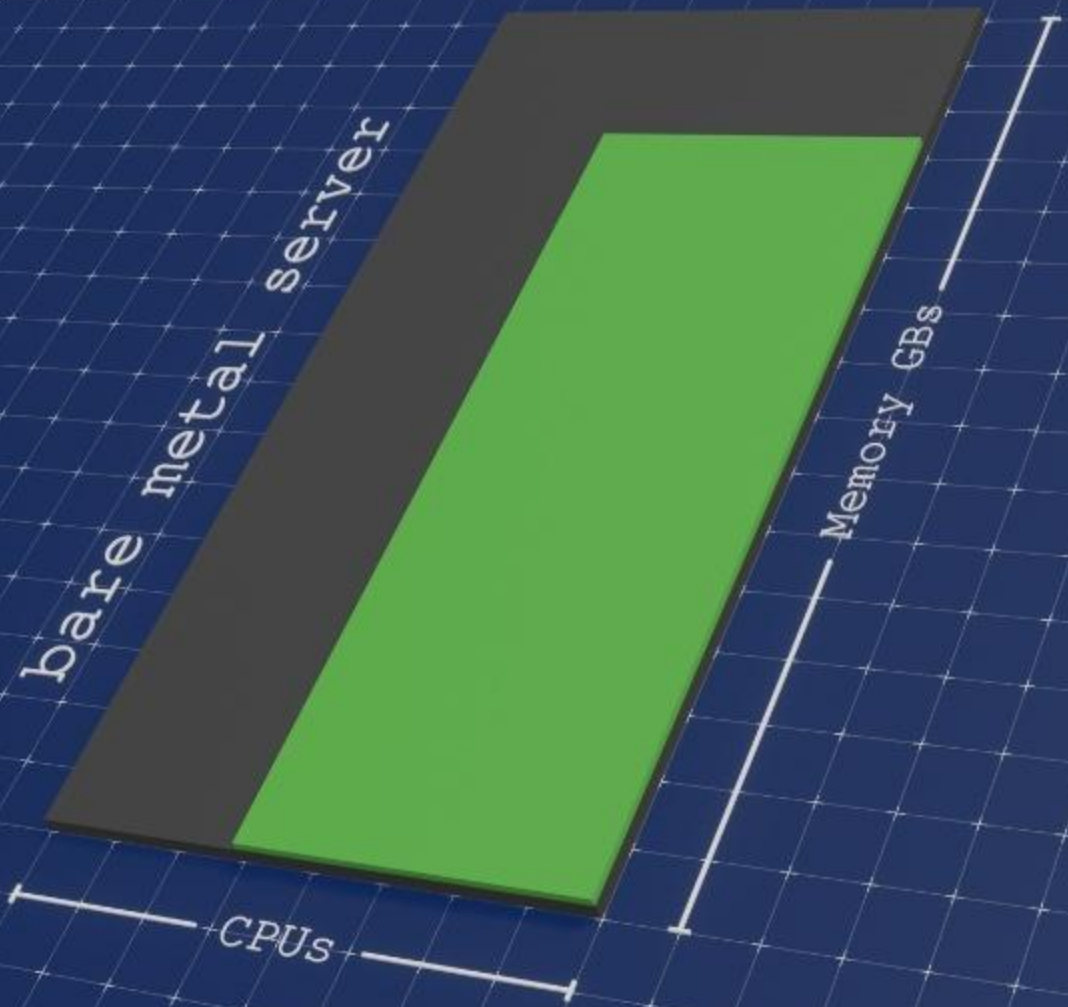

bare metal server
Memory GBs
CPUs

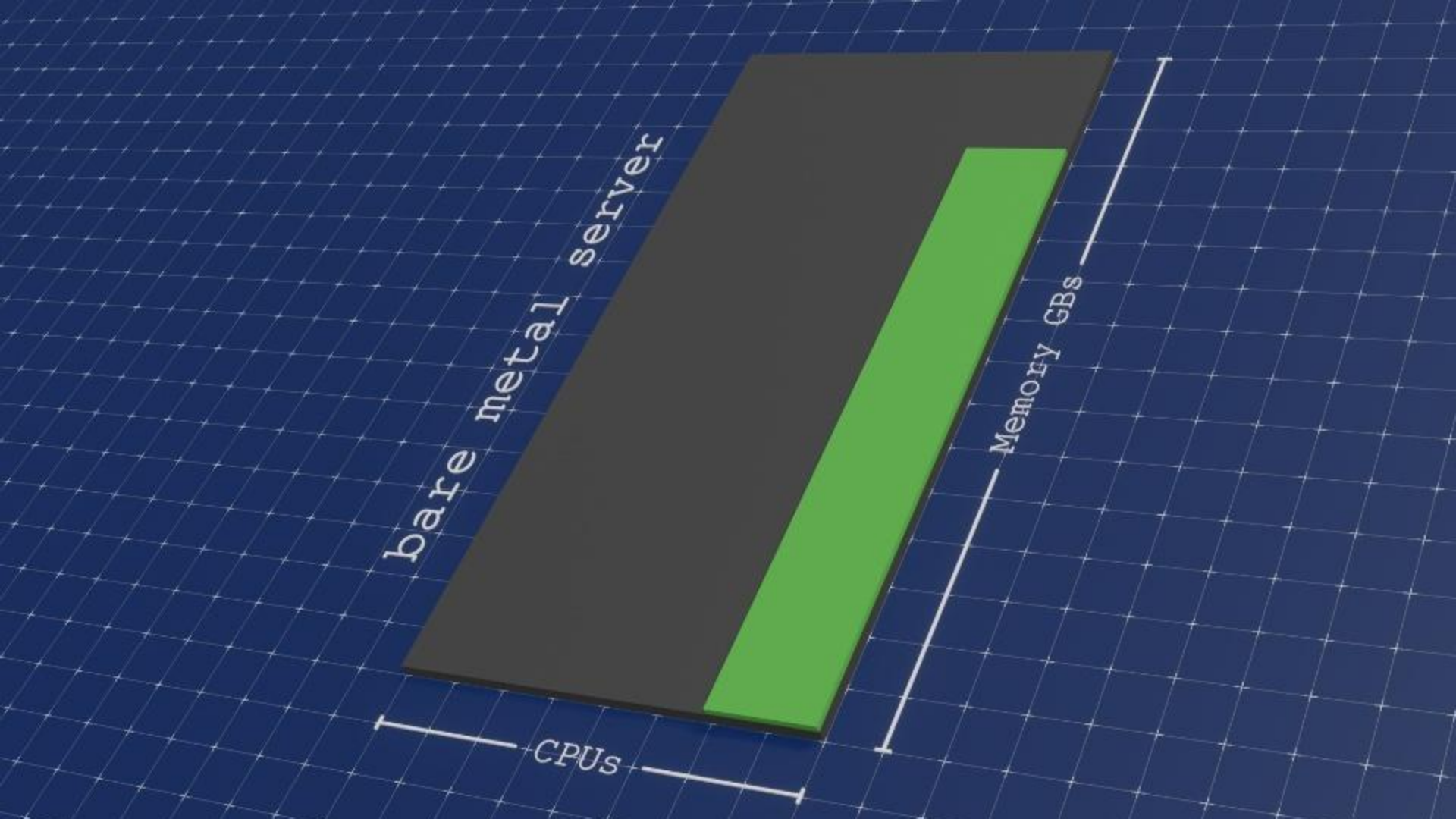

bare metal server
CPUs
Memory GBs

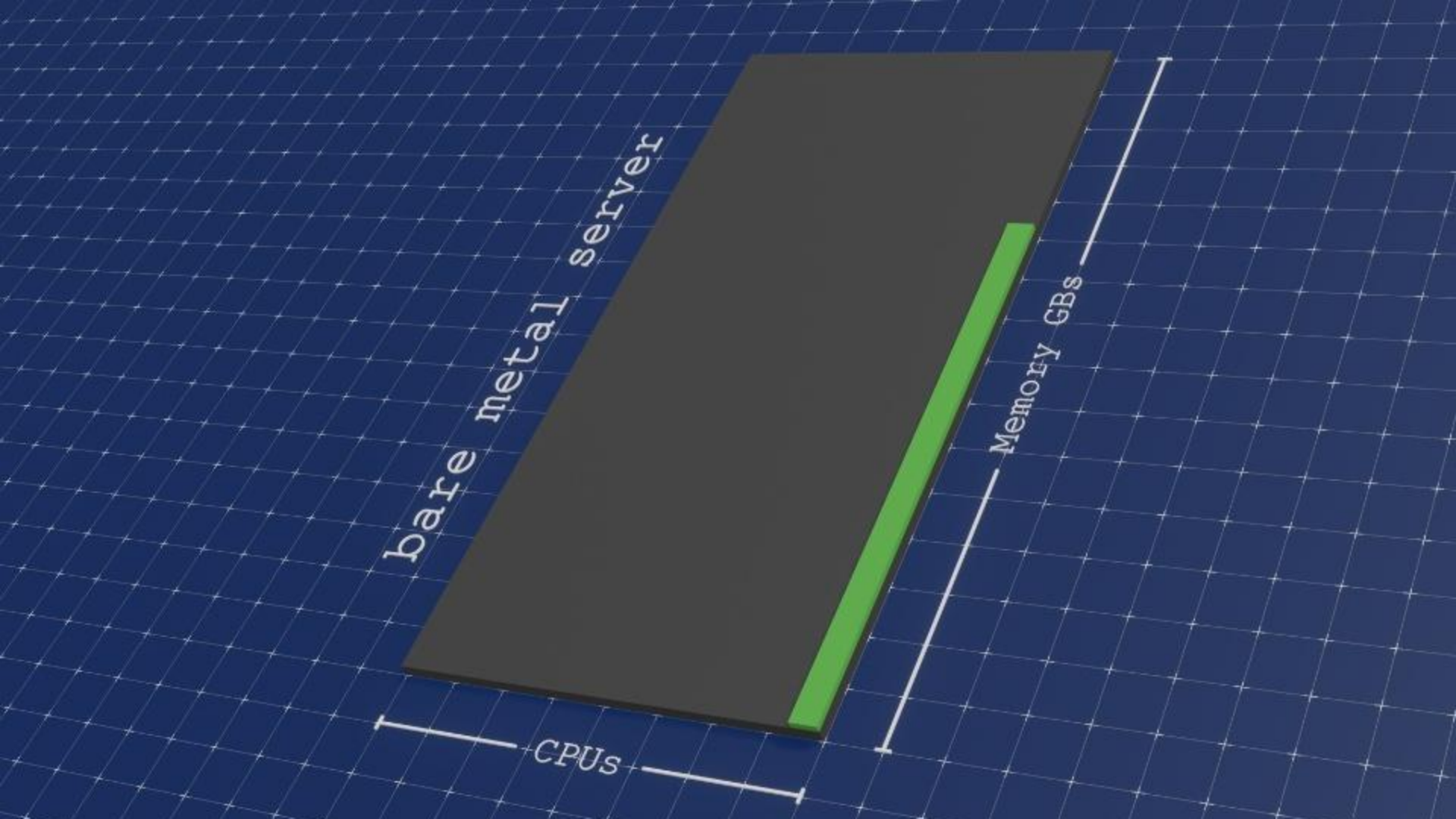

bare metal server
Memory GBs
CPUs

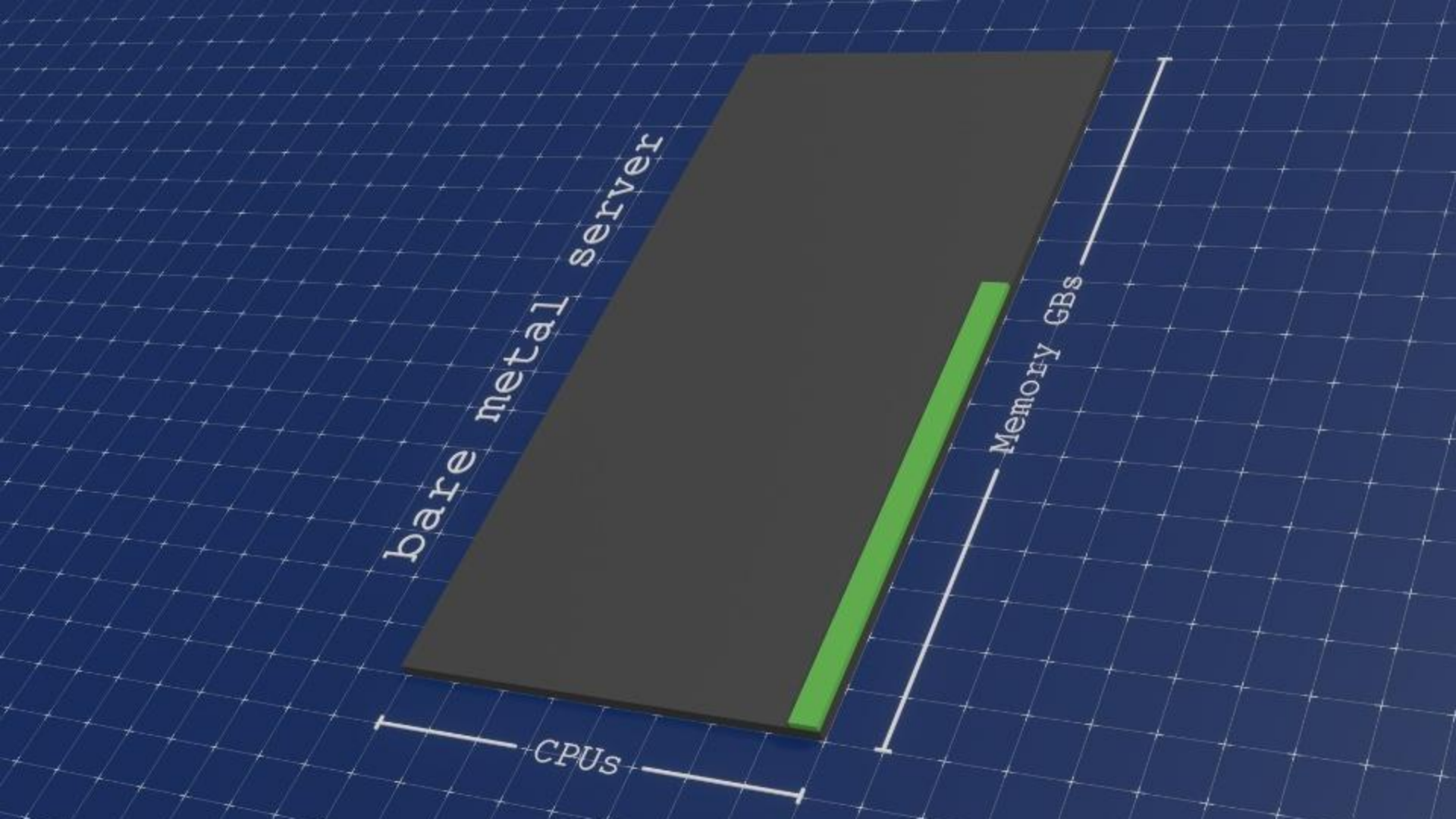

bare metal server
Memory GBs
CPUs

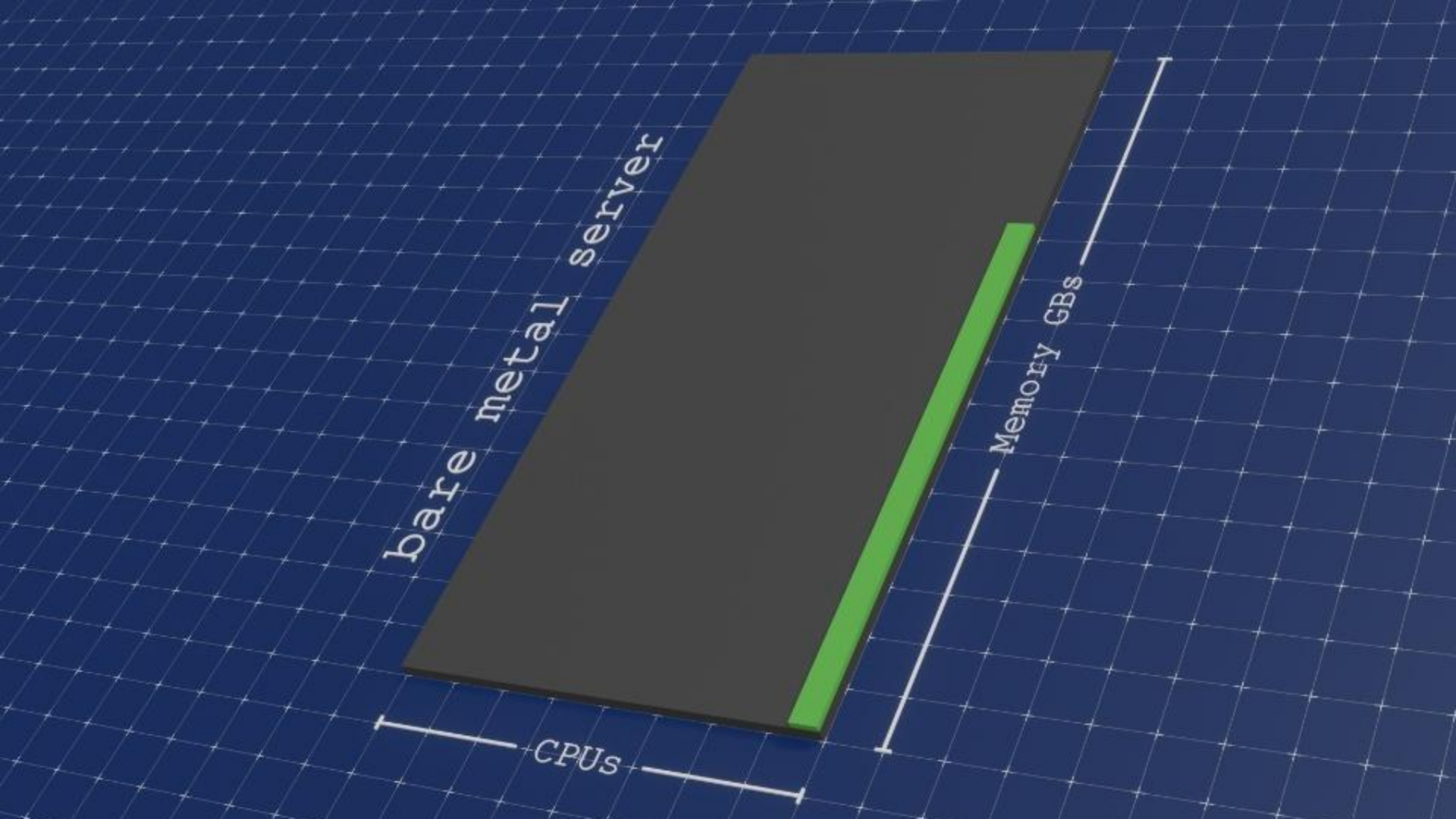

bare metal server
Memory GBs
CPUs

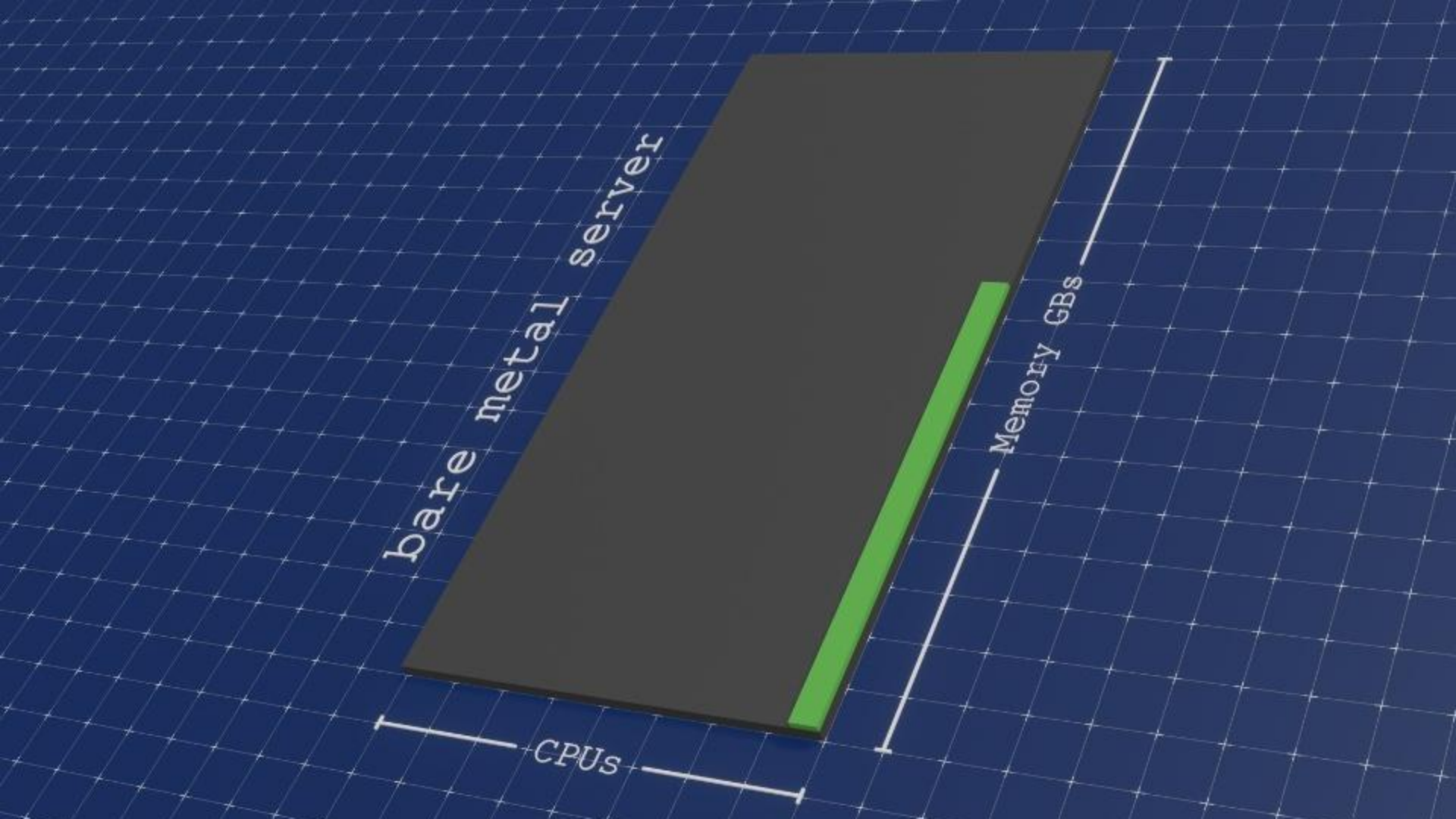

bare metal server
Memory GBs
CPUs

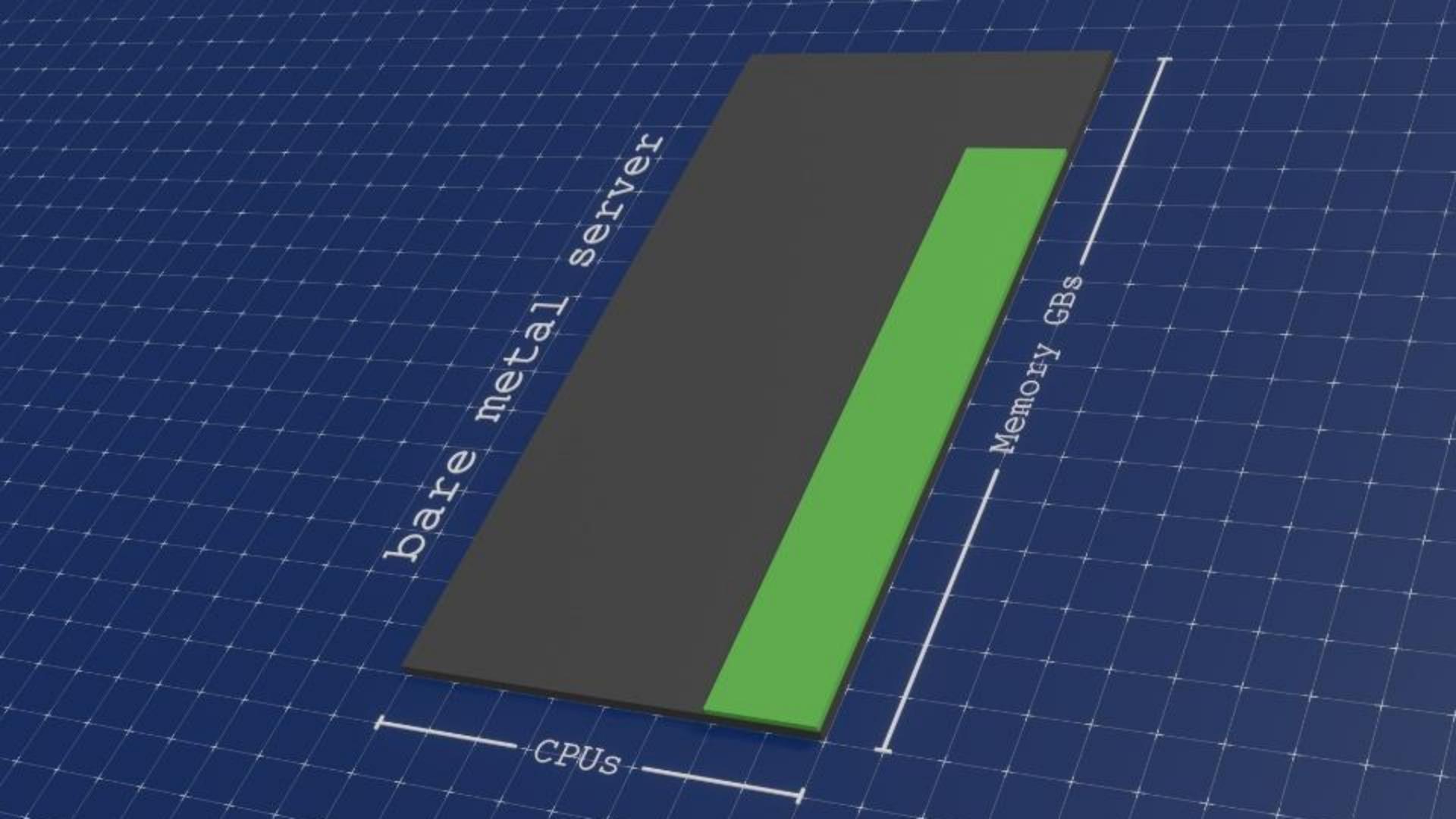

bare metal server
CPUs
Memory GBs

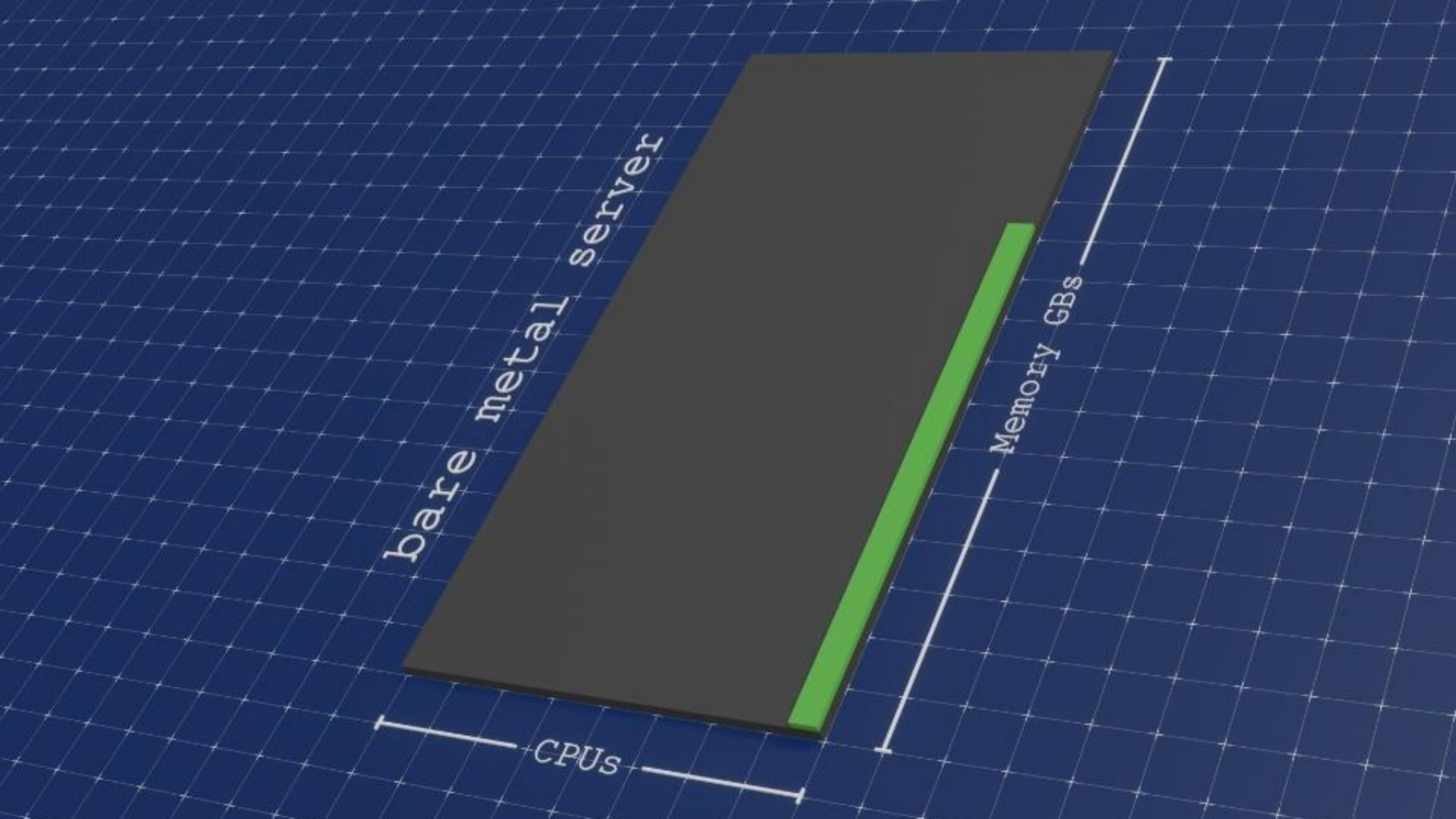

bare metal server
Memory GBs
CPUs

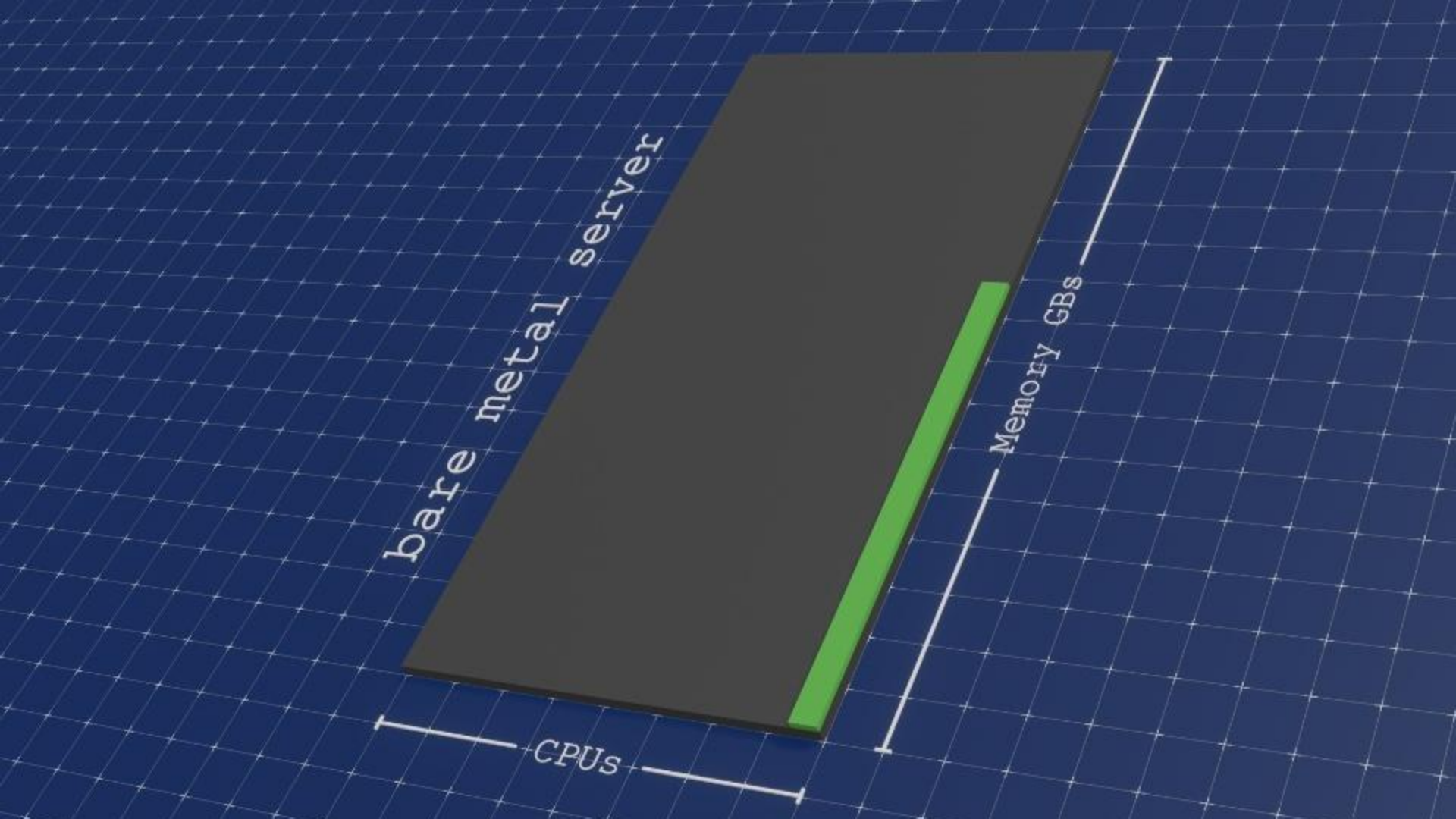

bare metal server
Memory GBs
CPUs

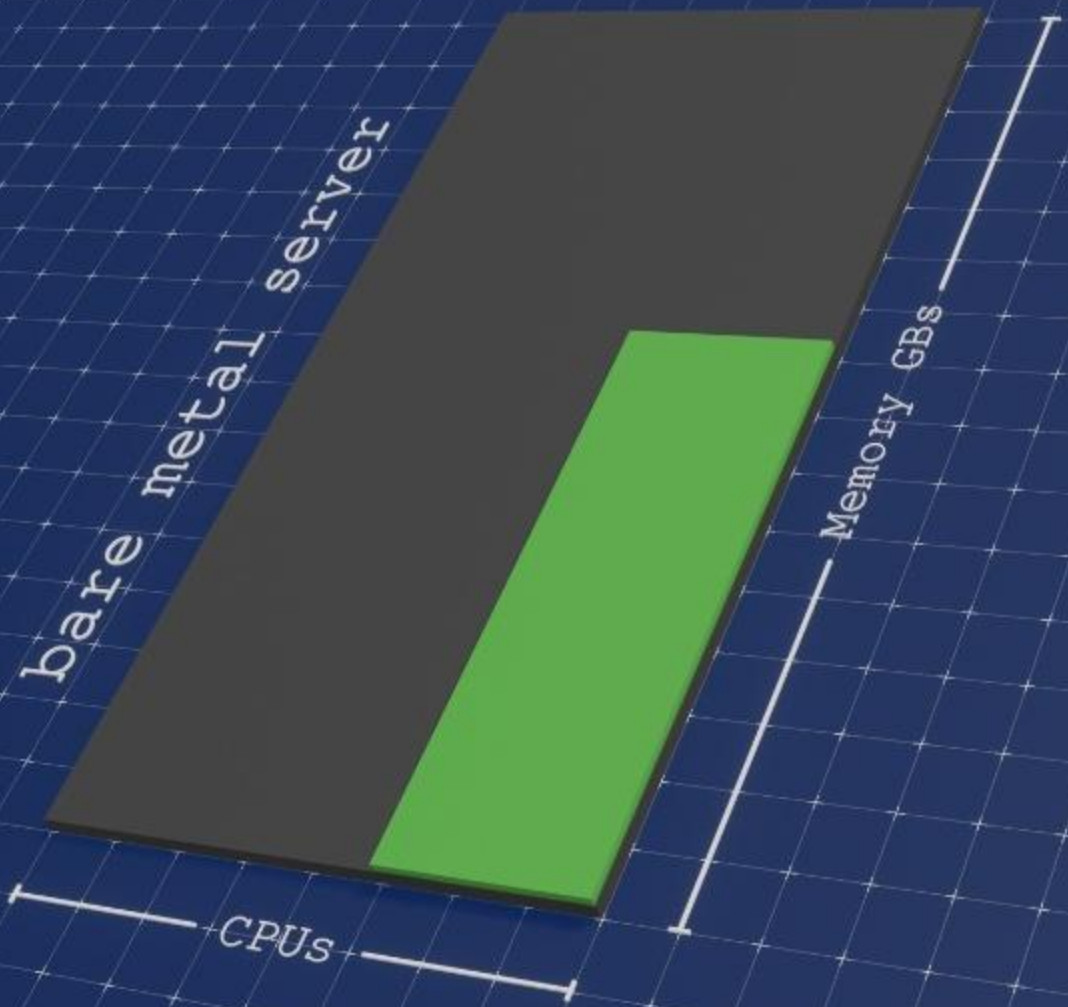

bare metal server
Memory GBs
CPUs

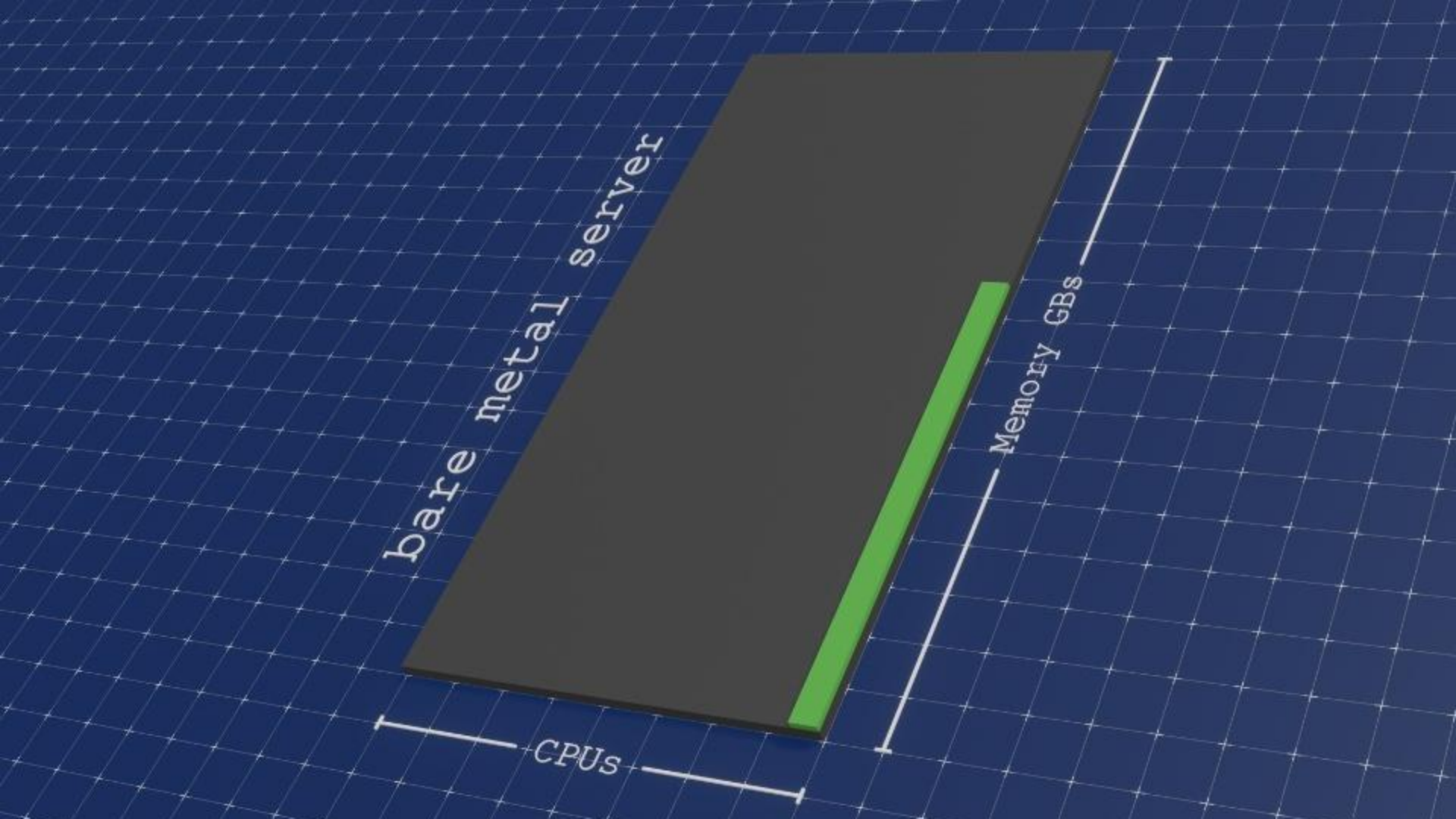

bare metal server
Memory GBs
CPUs

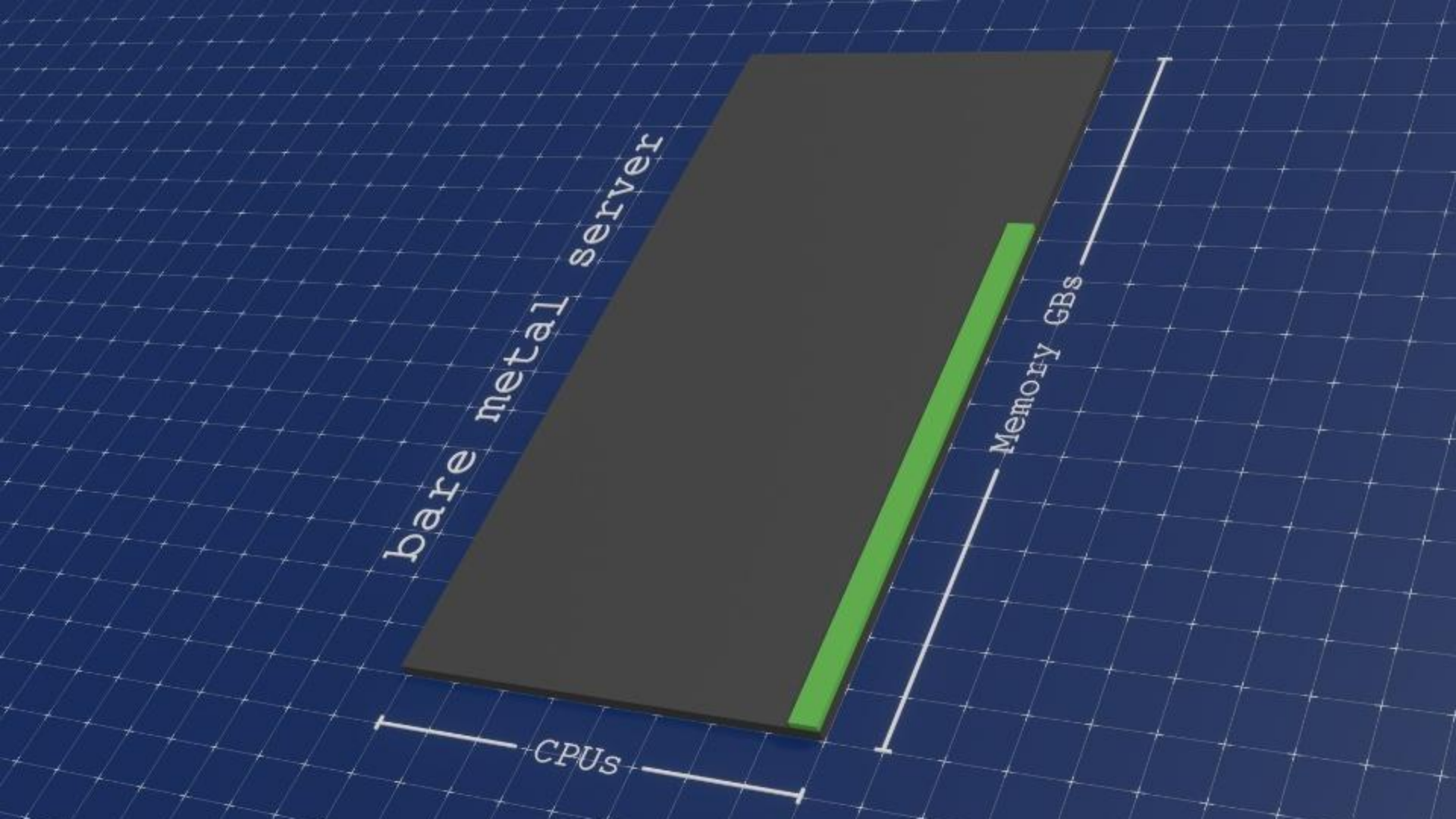

bare metal server
CPUs
Memory GBs

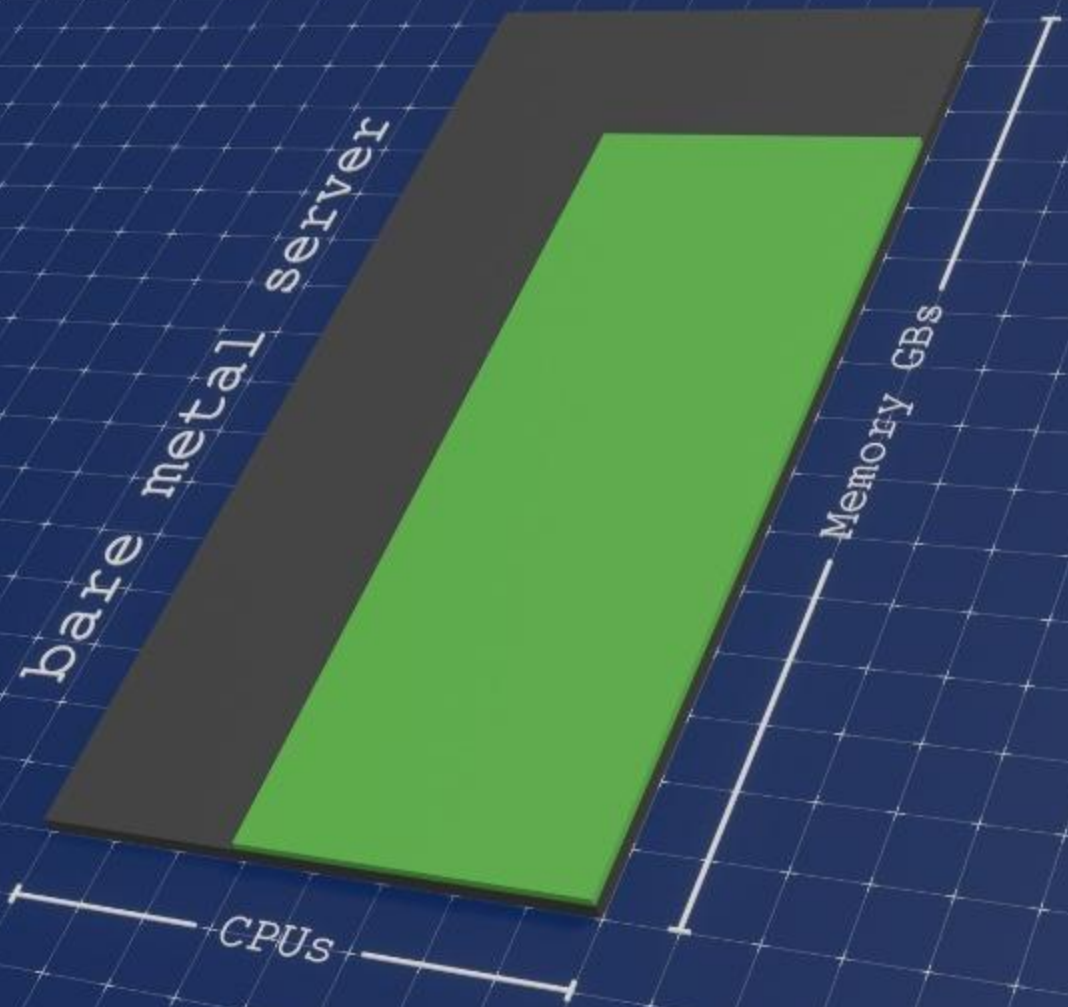

bare metal server
Memory GBs
CPUs

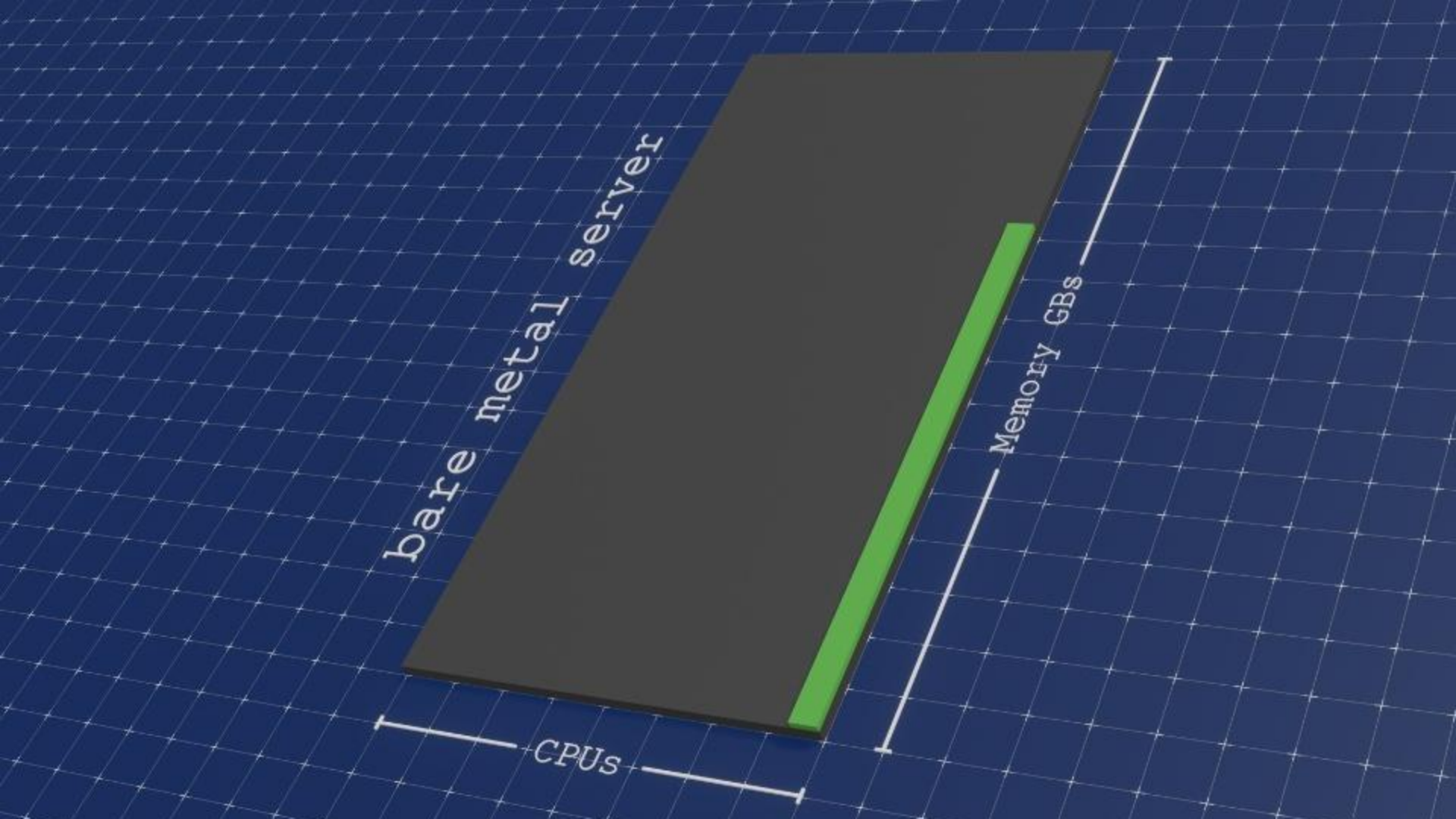

bare metal server
Memory GBs
CPUs

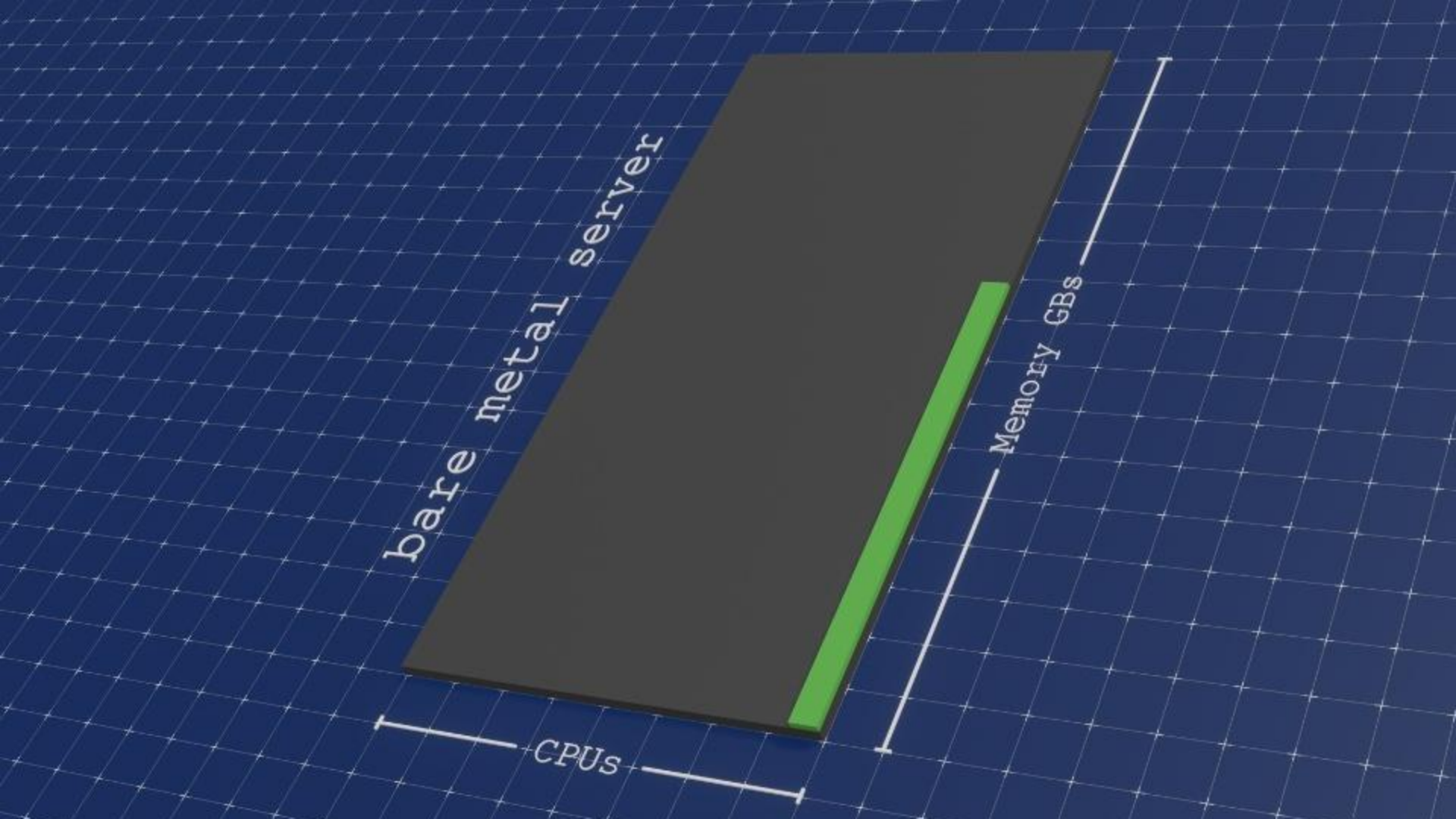

bare metal server
Memory GBs
CPUs

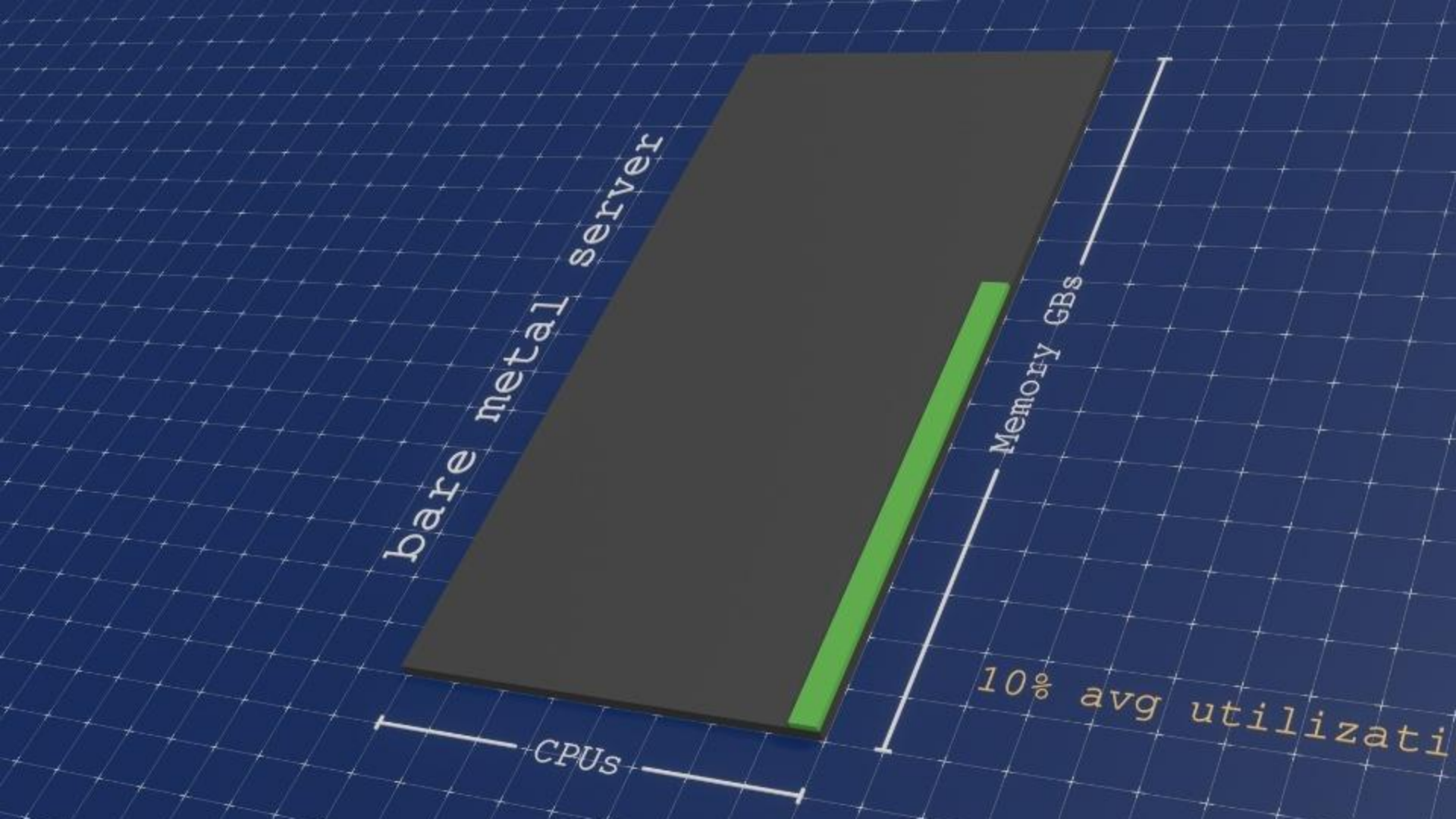

bare metal server
Memory GBs
CPUs

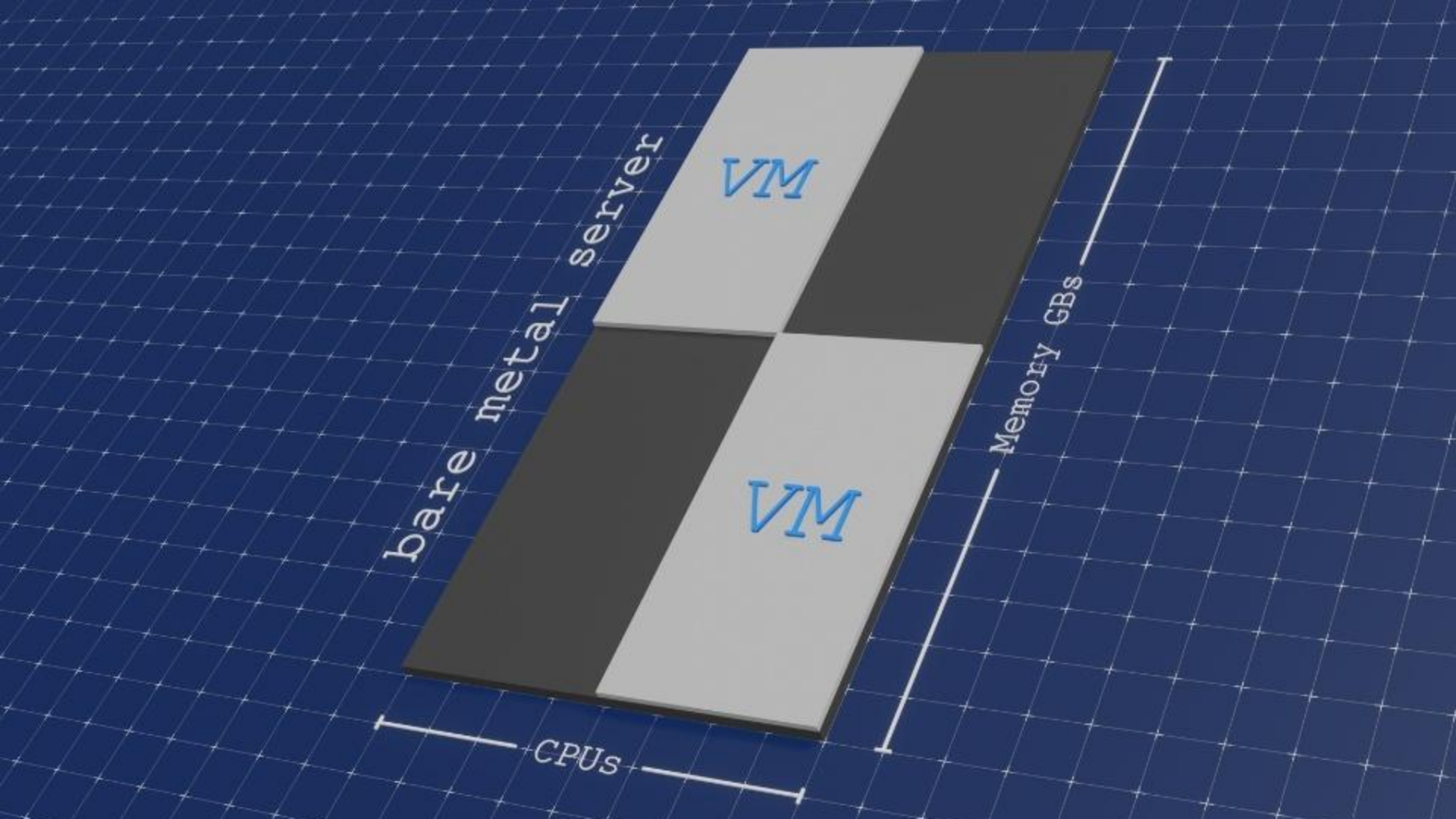

bare metal server
Memory GBs
CPUs

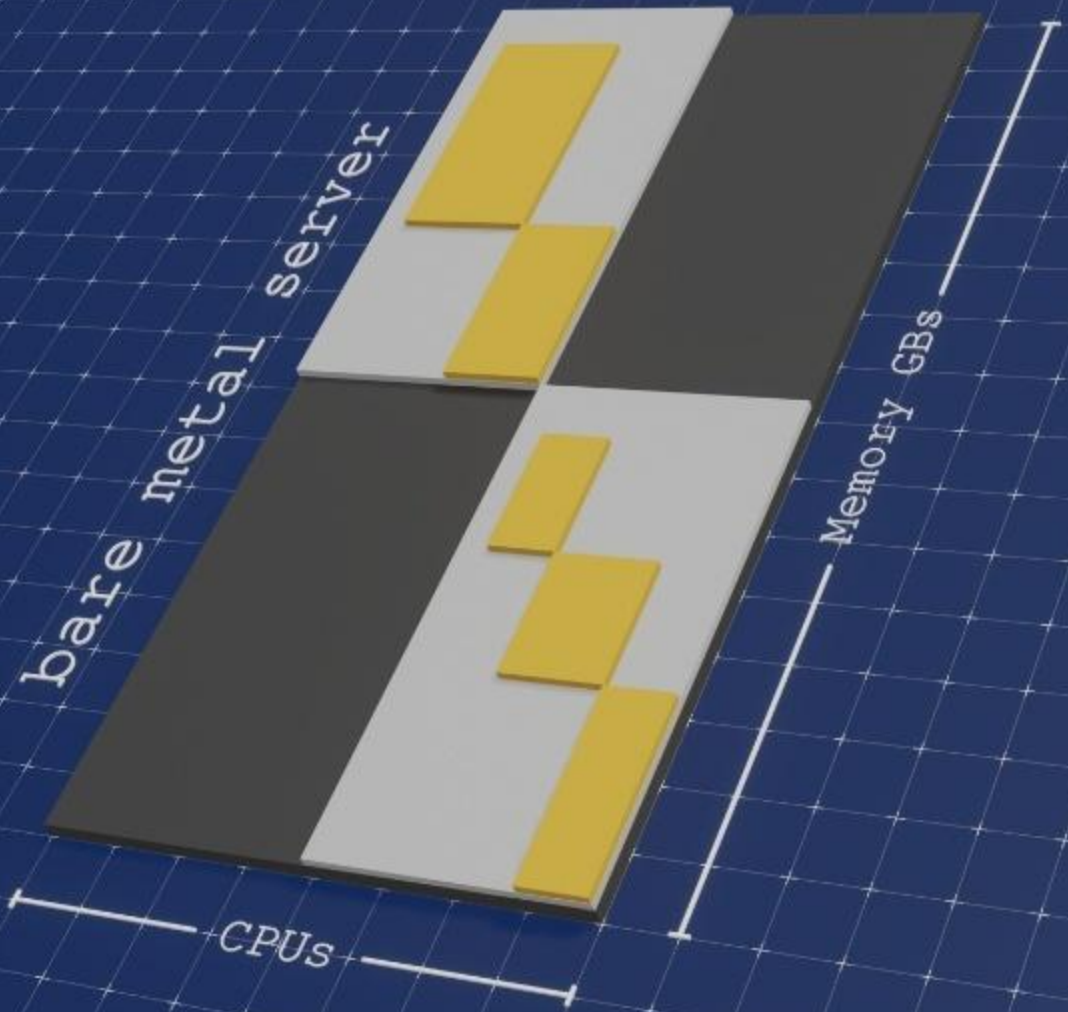

bare metal server
Memory GBs
CPUs

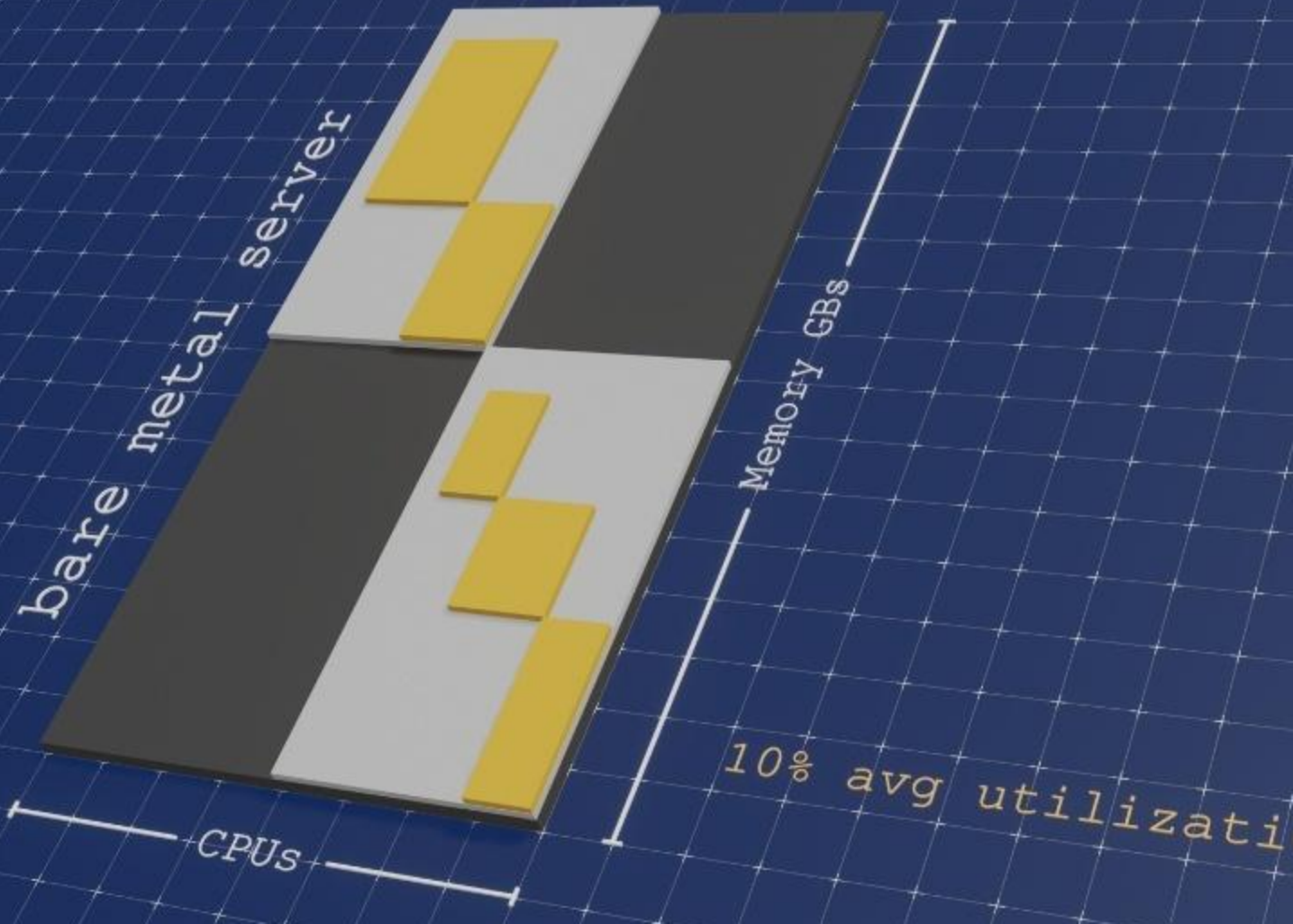

bare metal server
Memory GBs
CPUs

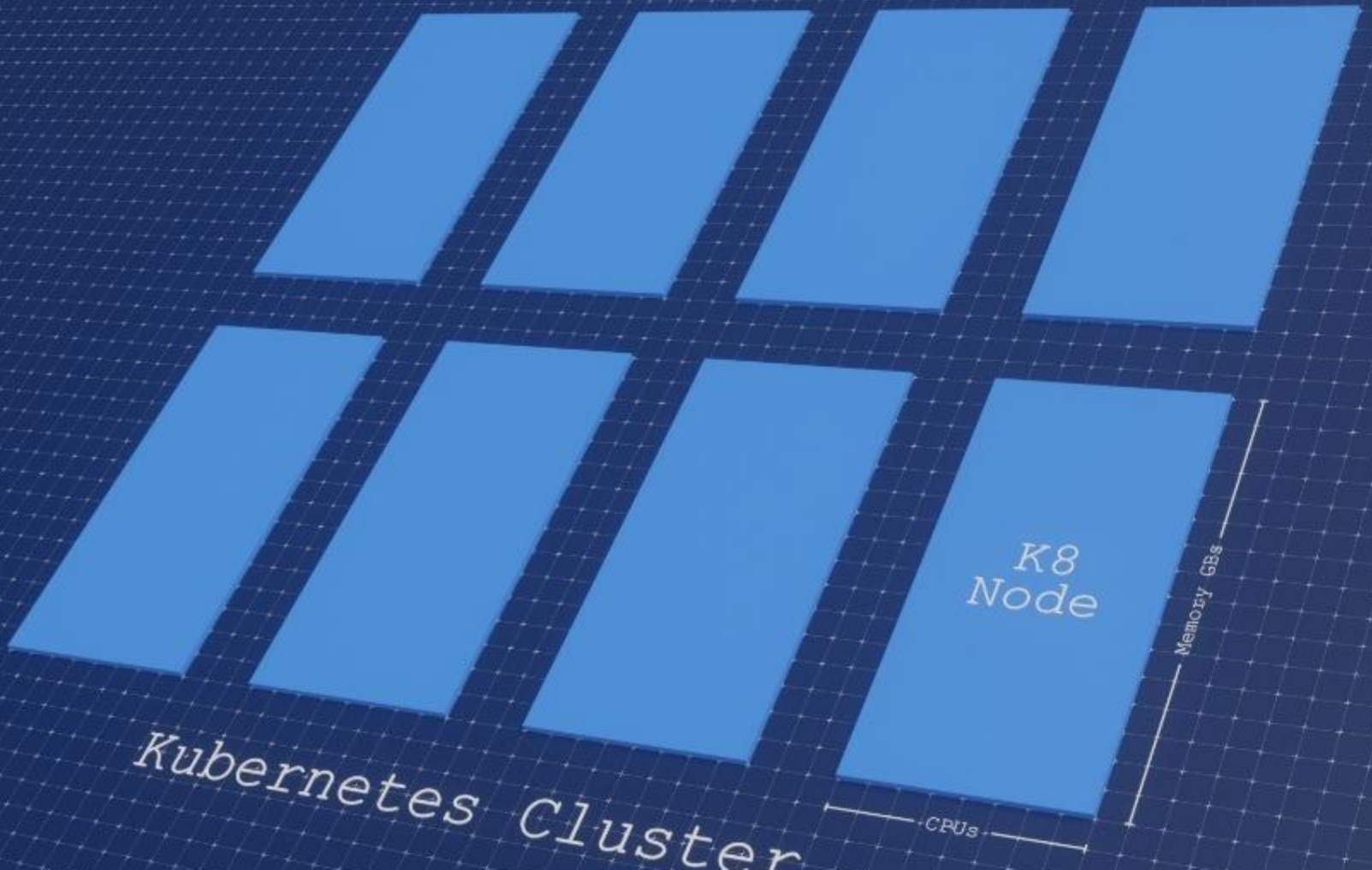

bare metal server
Memory GBs
CPUs

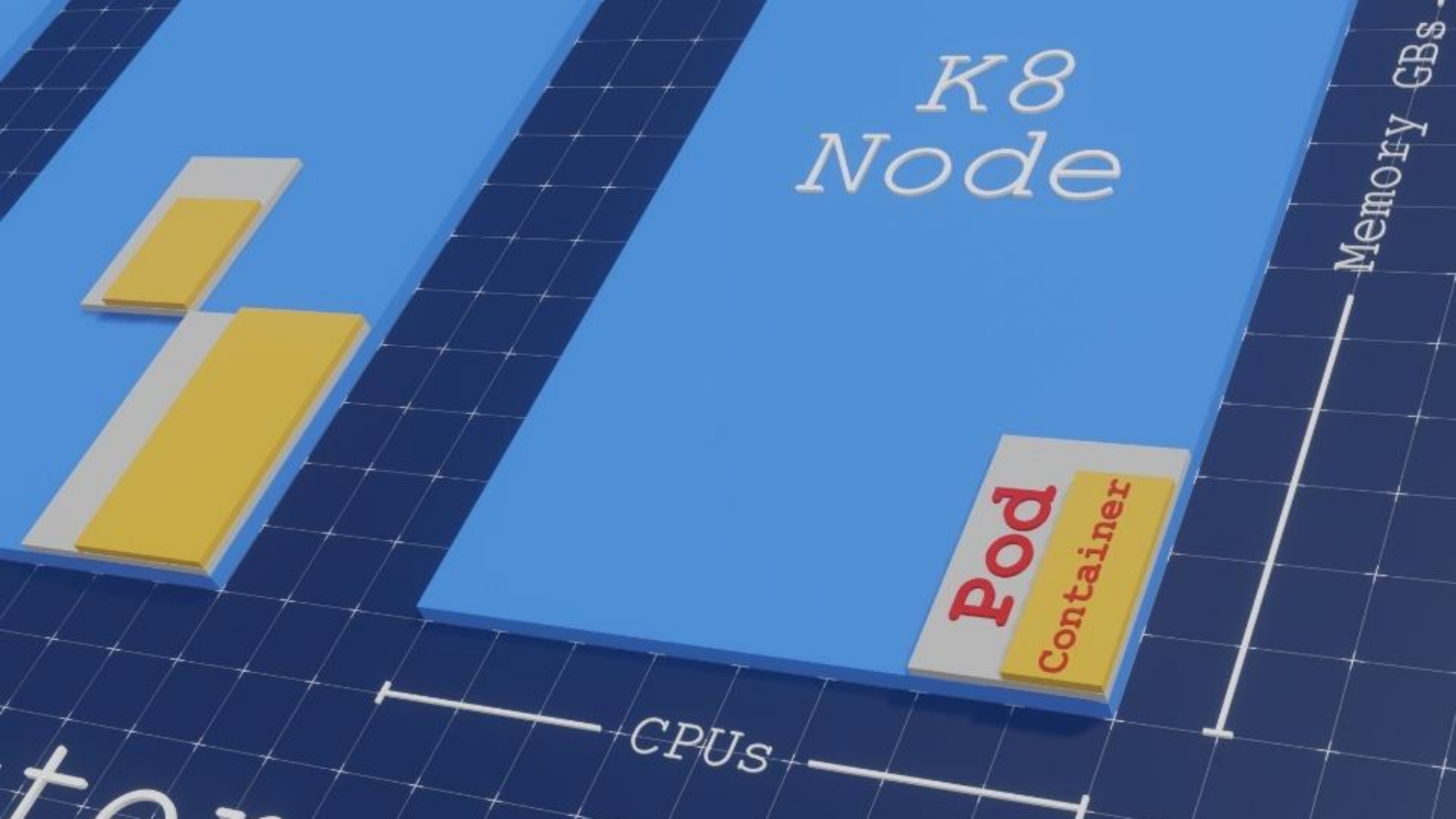

bare metal server
CPUs
Memory GBs
10% avg utilizati

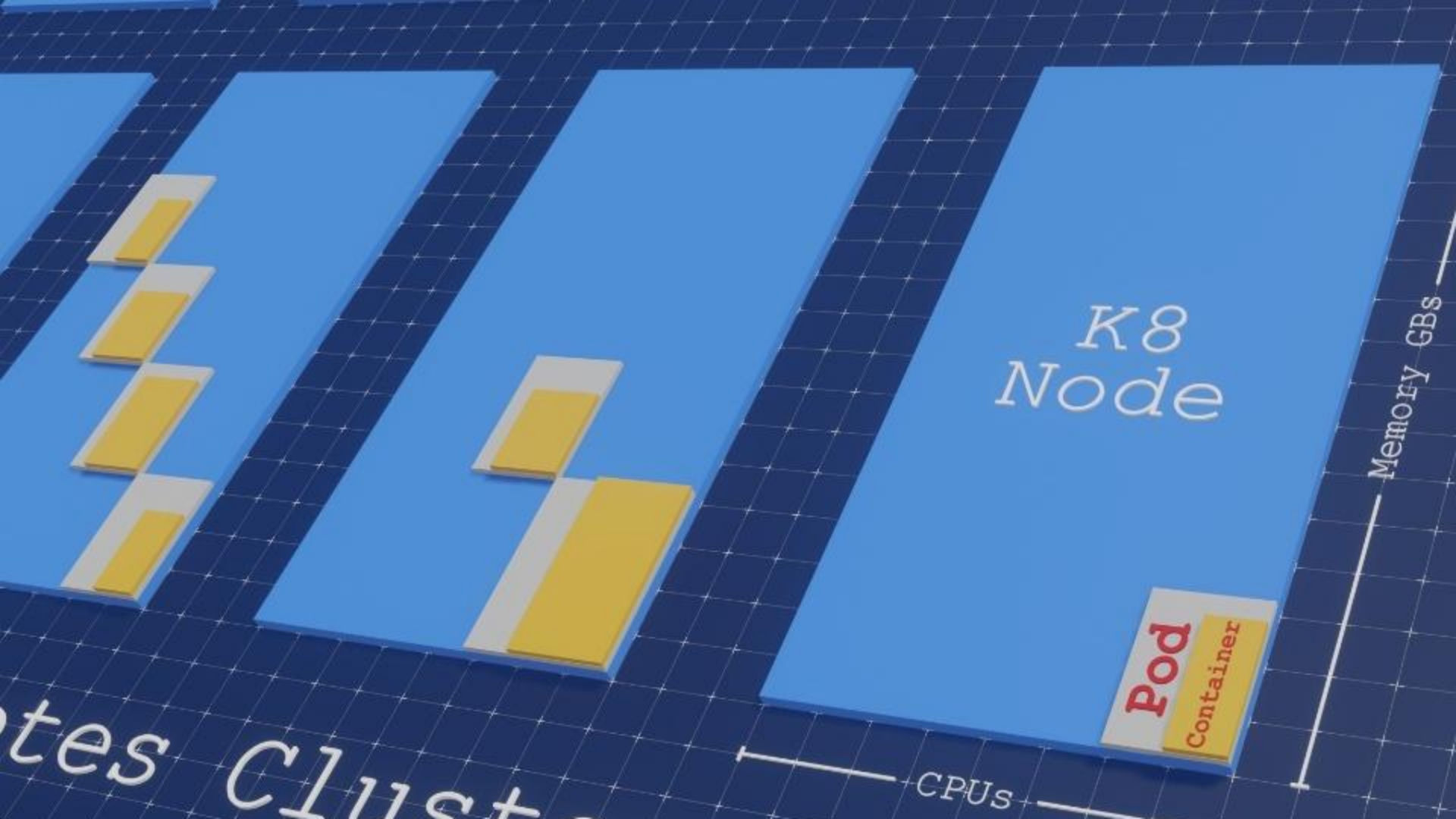

bare metal server
VM
VM
Memory GBs
CPUs

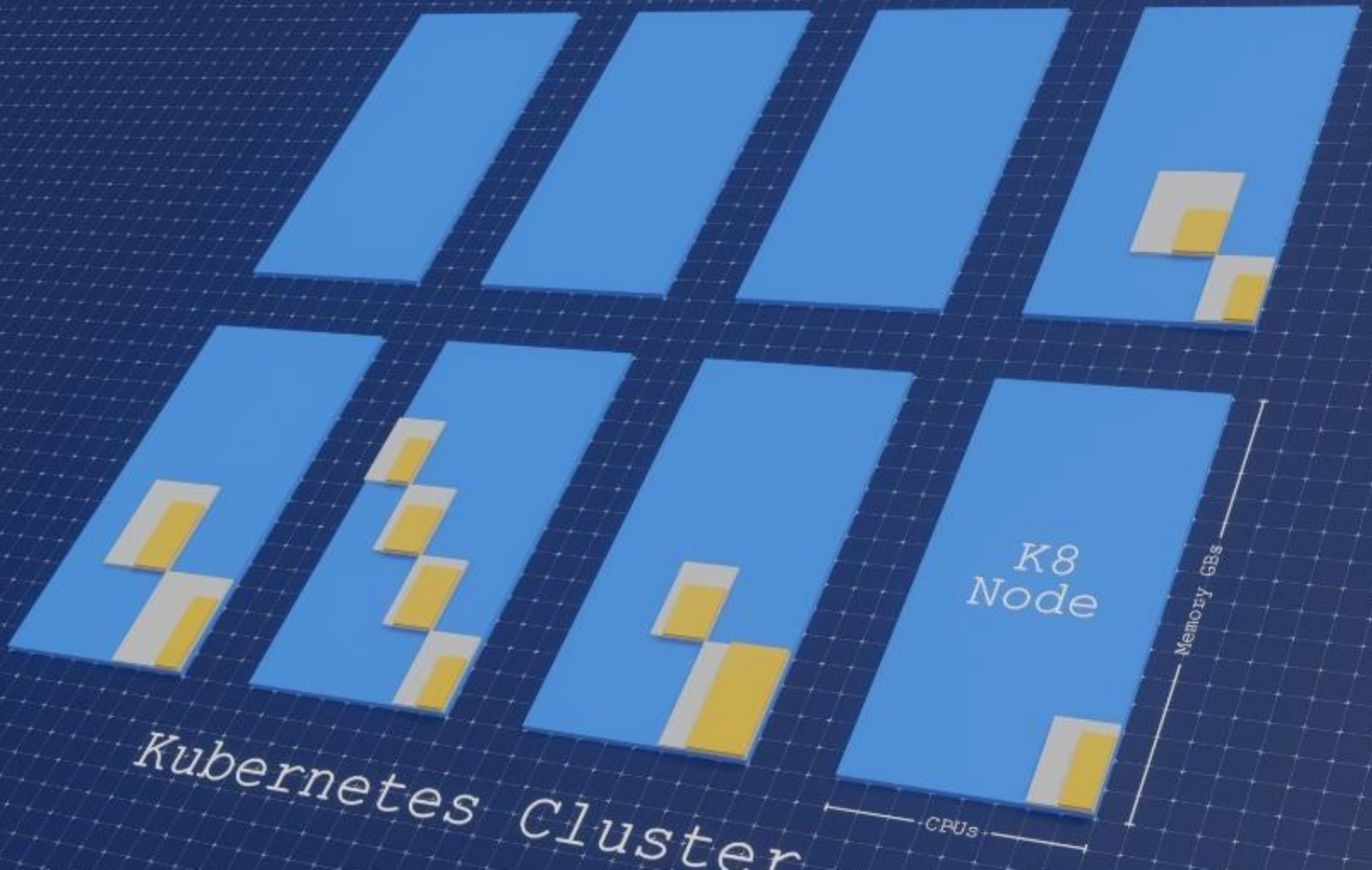

bare metal server
CPUs
Memory GBs

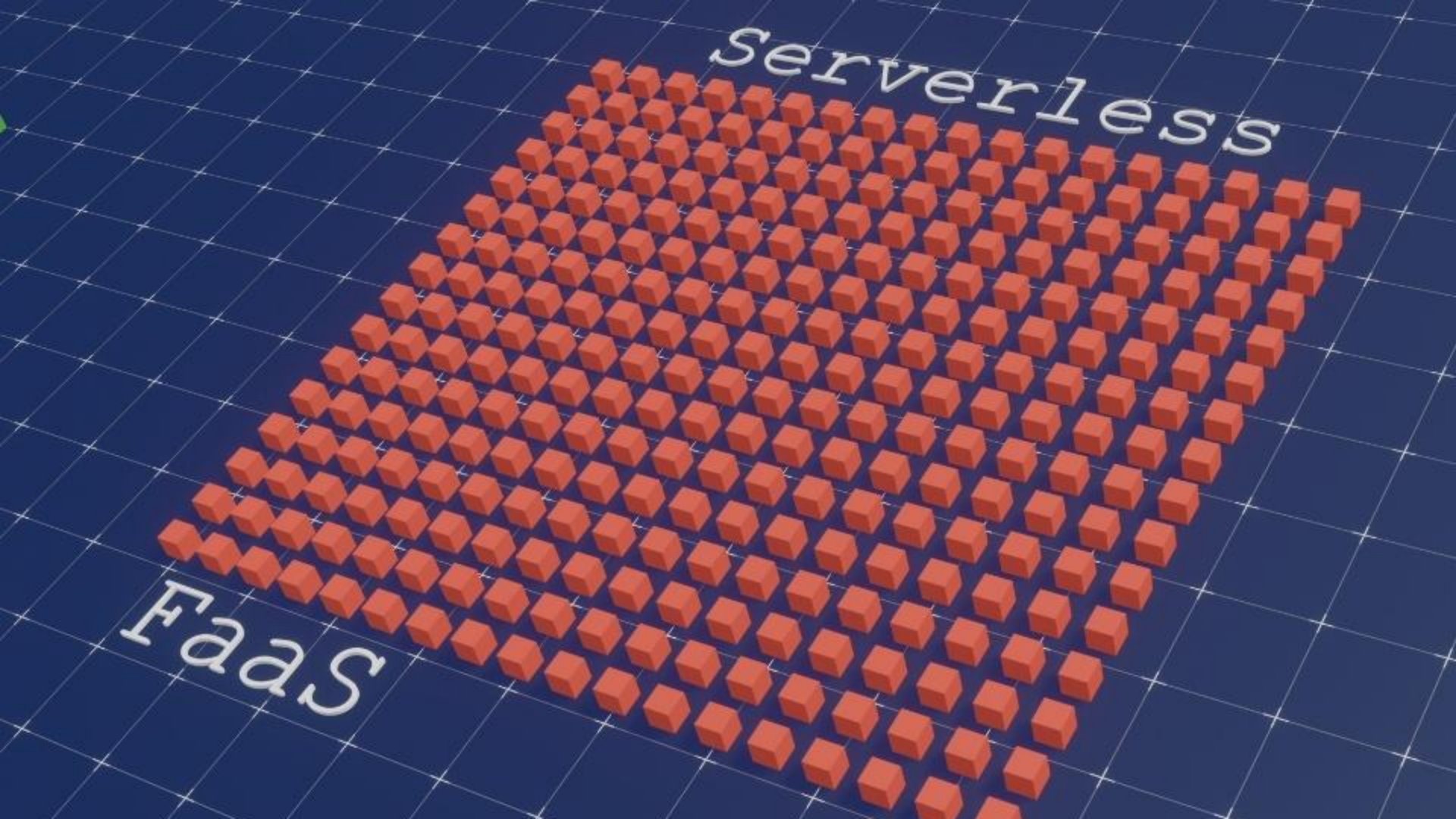

bare metal server
Memory GBs
CPUs
10% avg utilizati

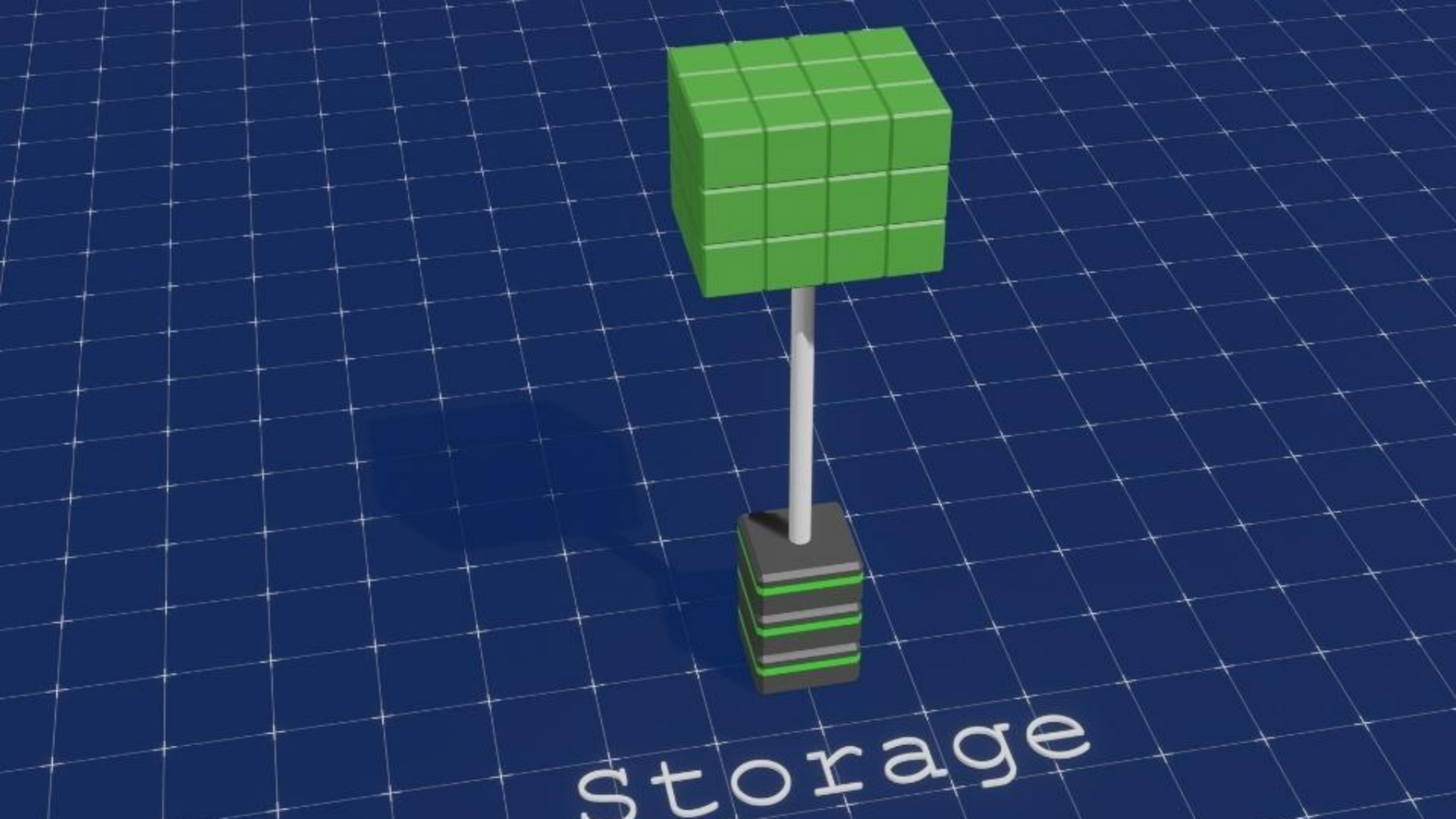

K8
Node
Memory GBs
CPUs
Kubernetes Cluster

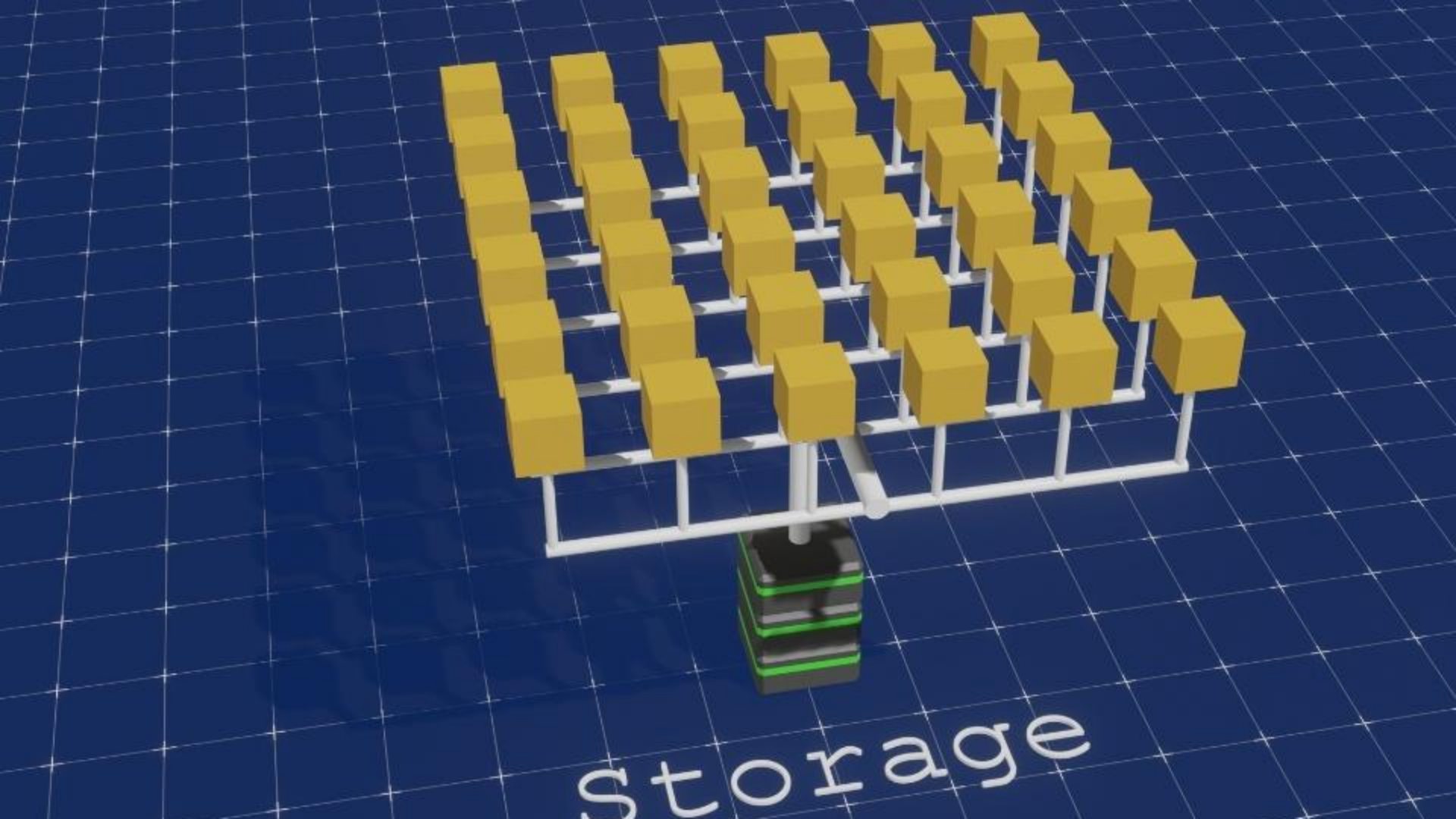

K8
Node
Pod
Container
CPUs
Memory GBs

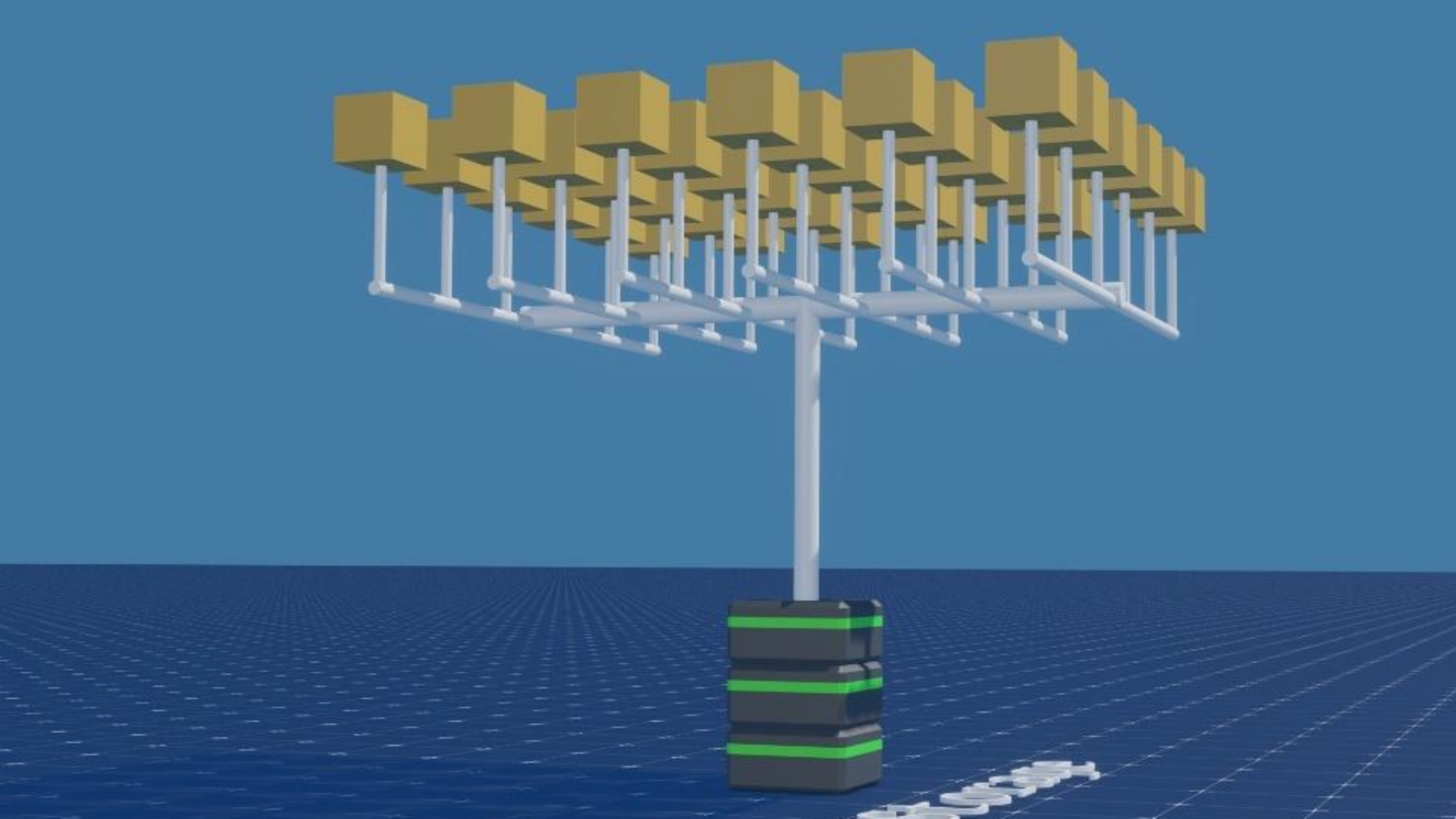

K8
Node
Pod
Container
CPUs
Memory GBs
tes Clust

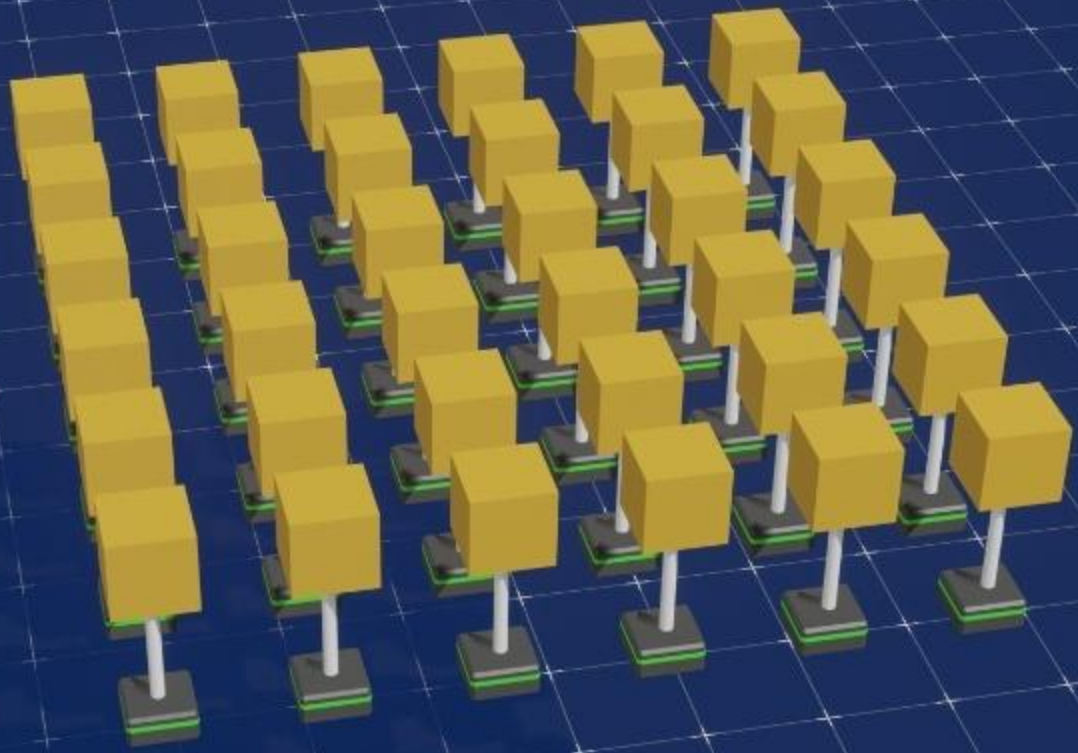

K8
Node
Memory GBs
CPUs
Kubernetes Cluster


Serverless
FaaS

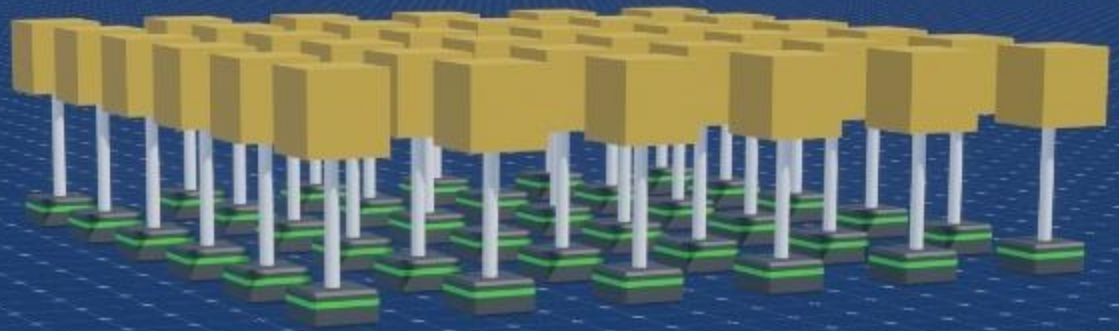

Storage


Storage

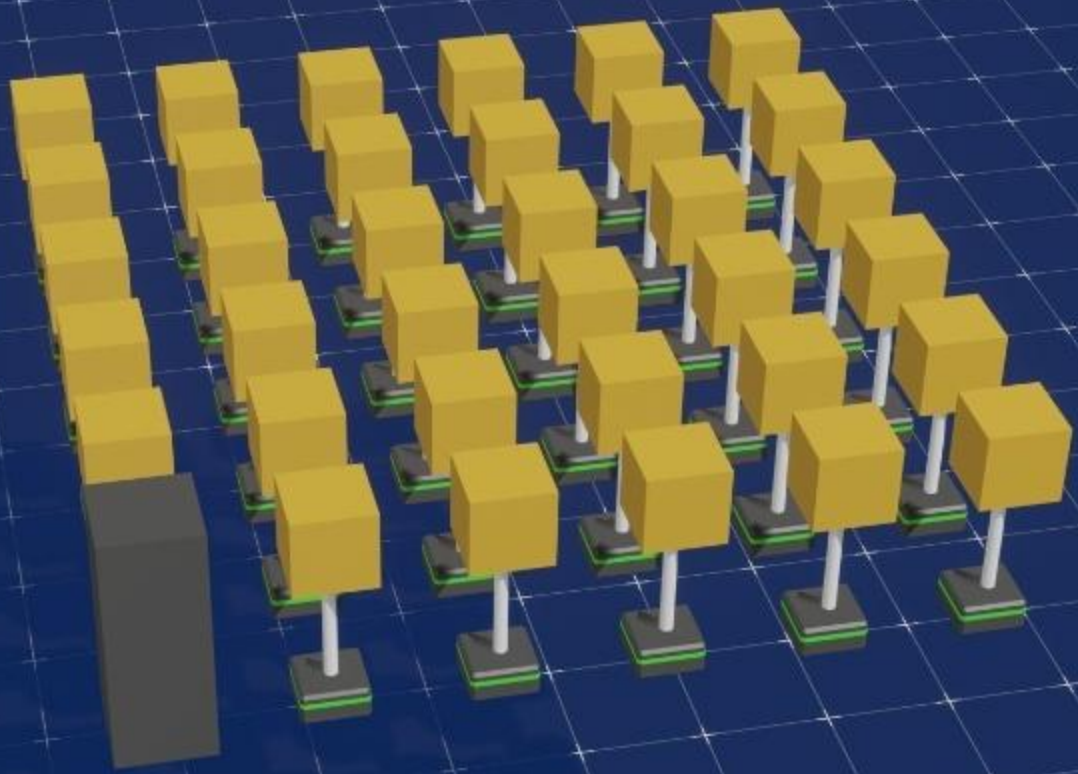




loosley coupled

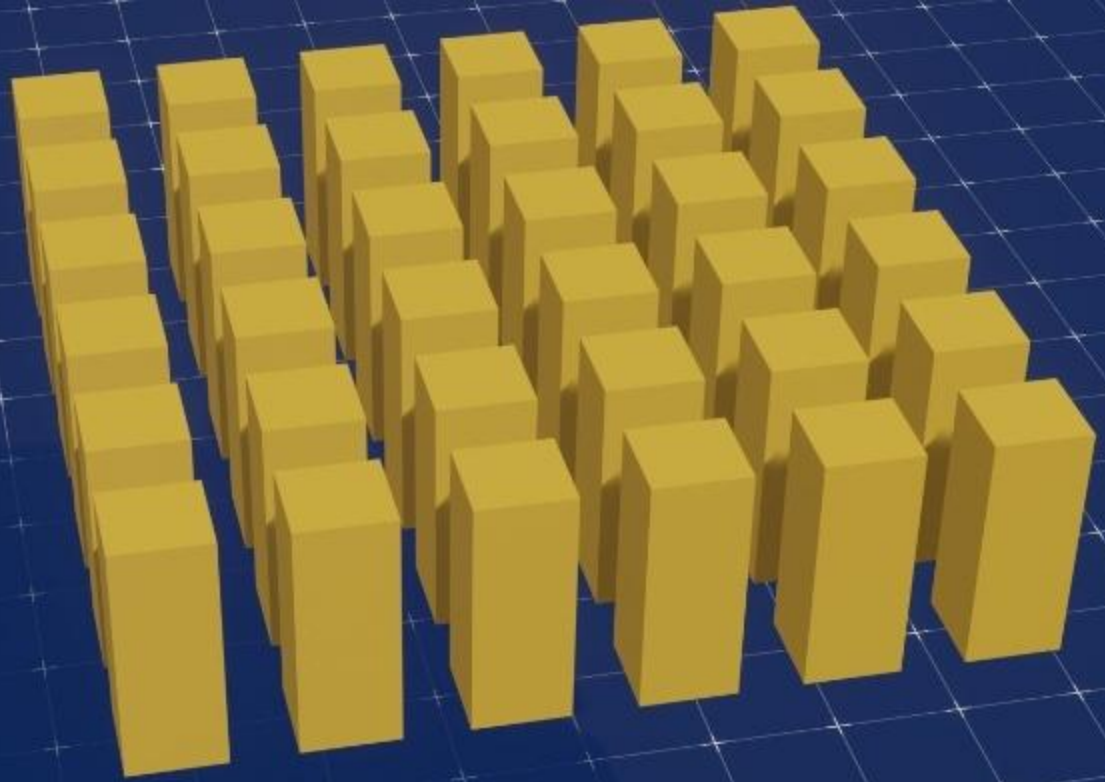


loosley coupled

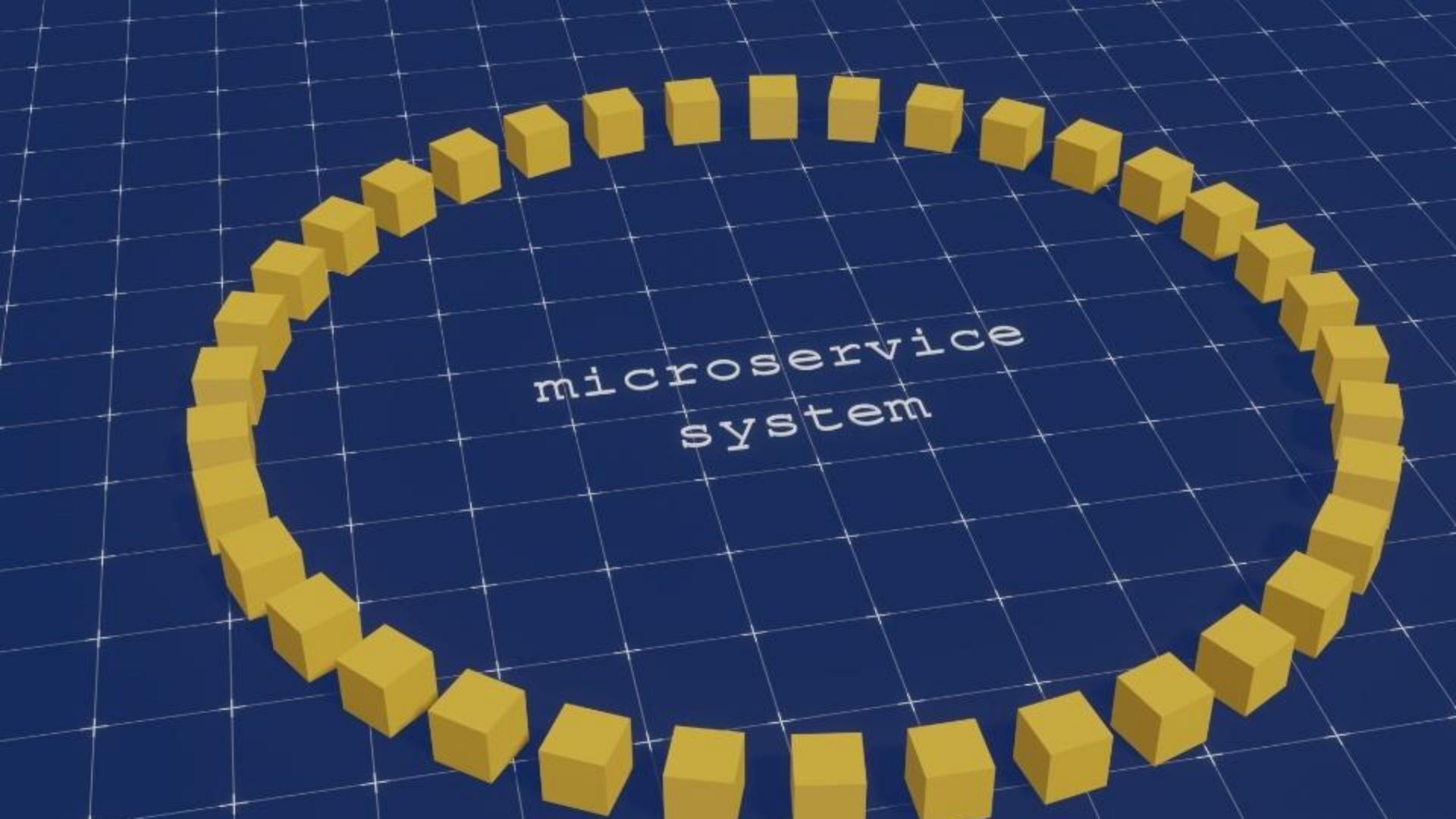


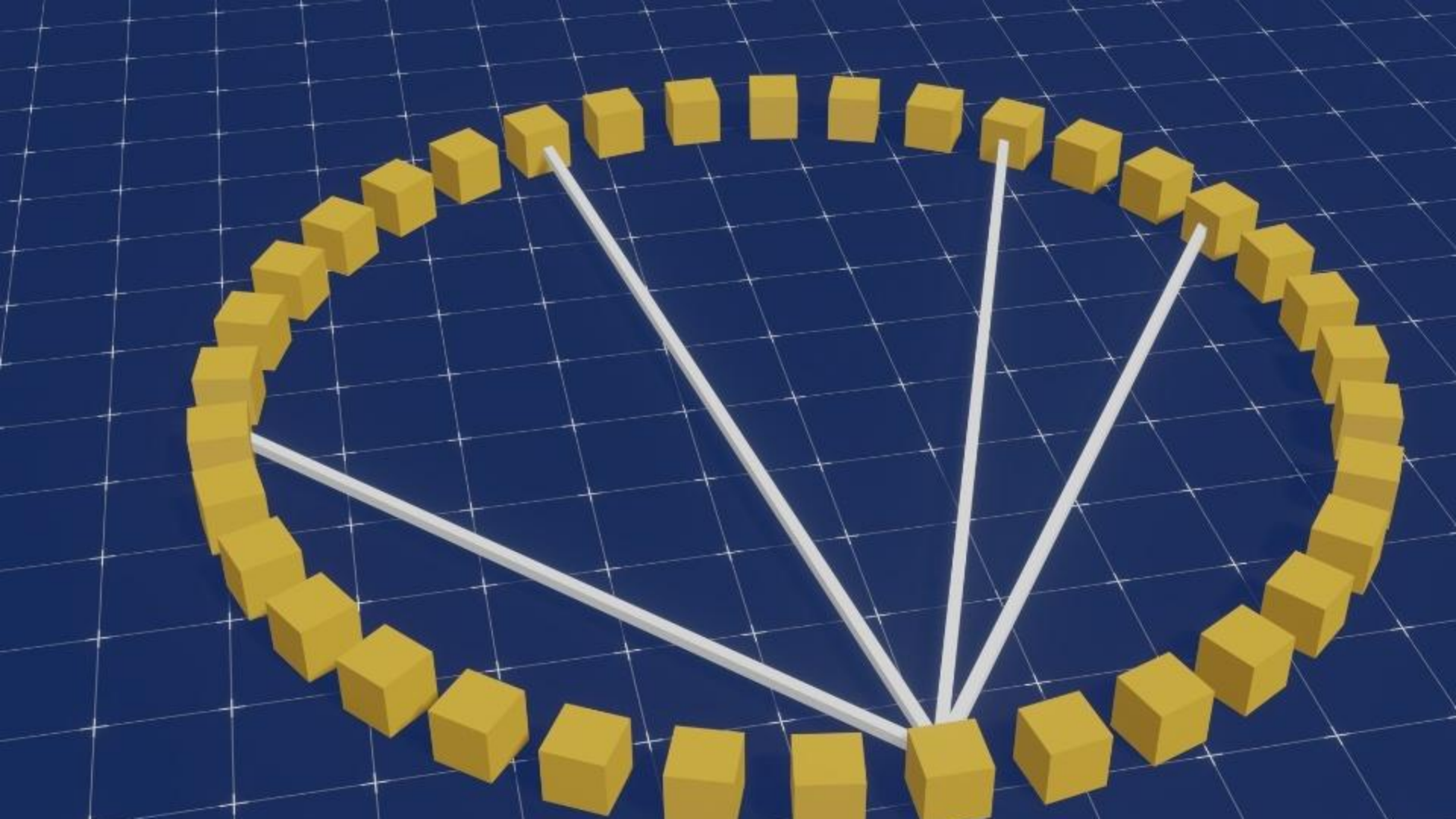
loosley coupled

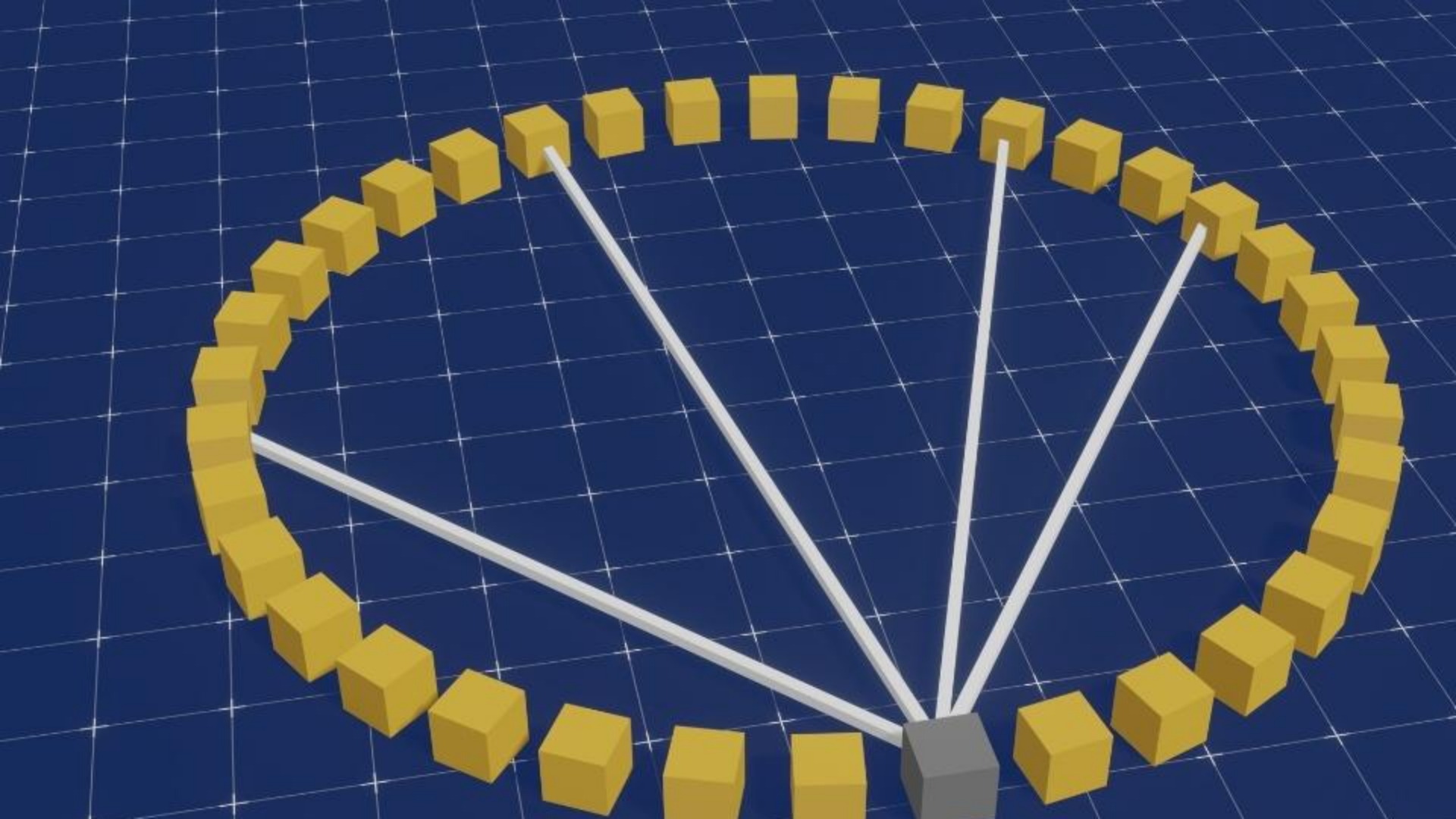
loosley coupled

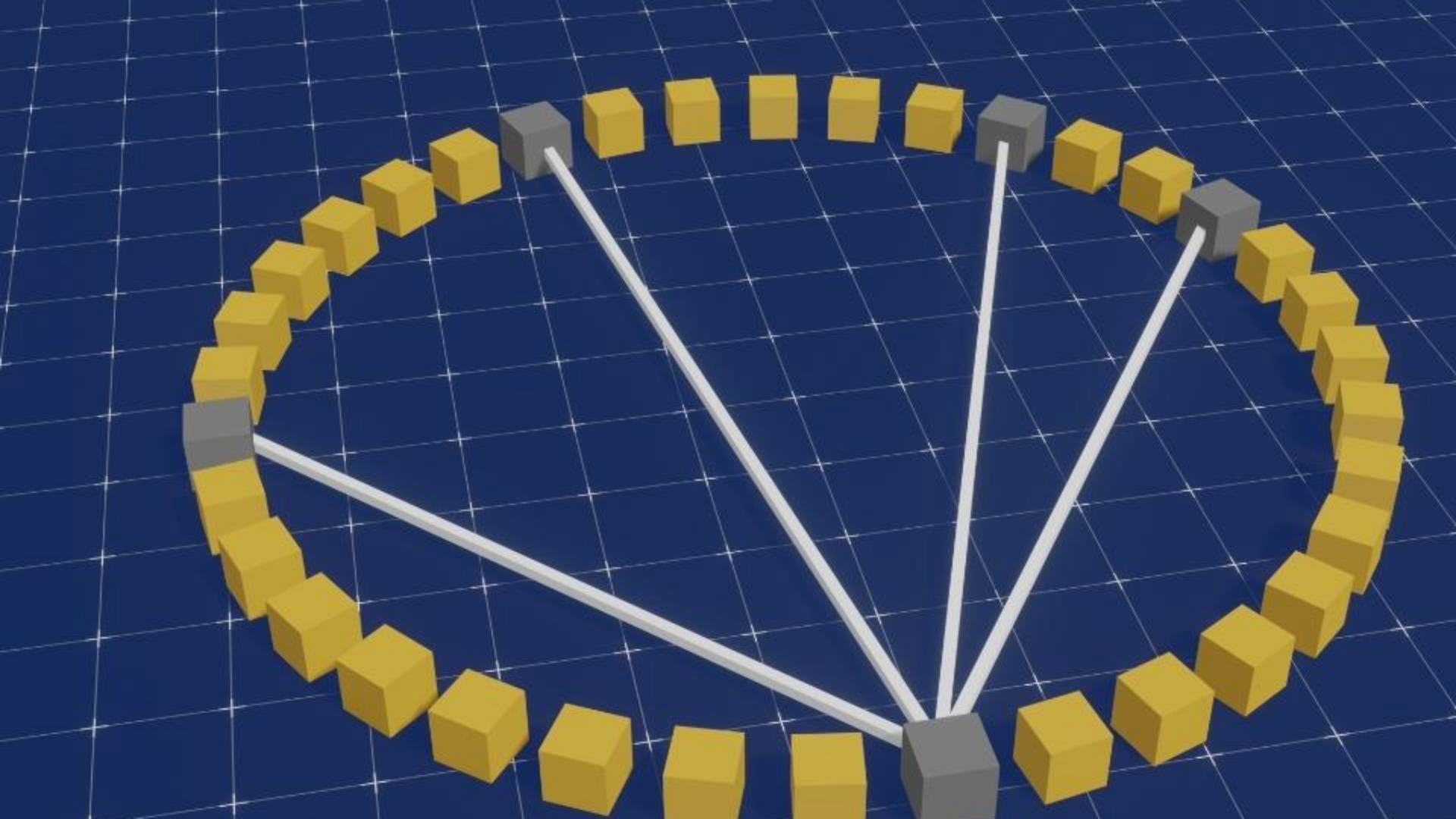


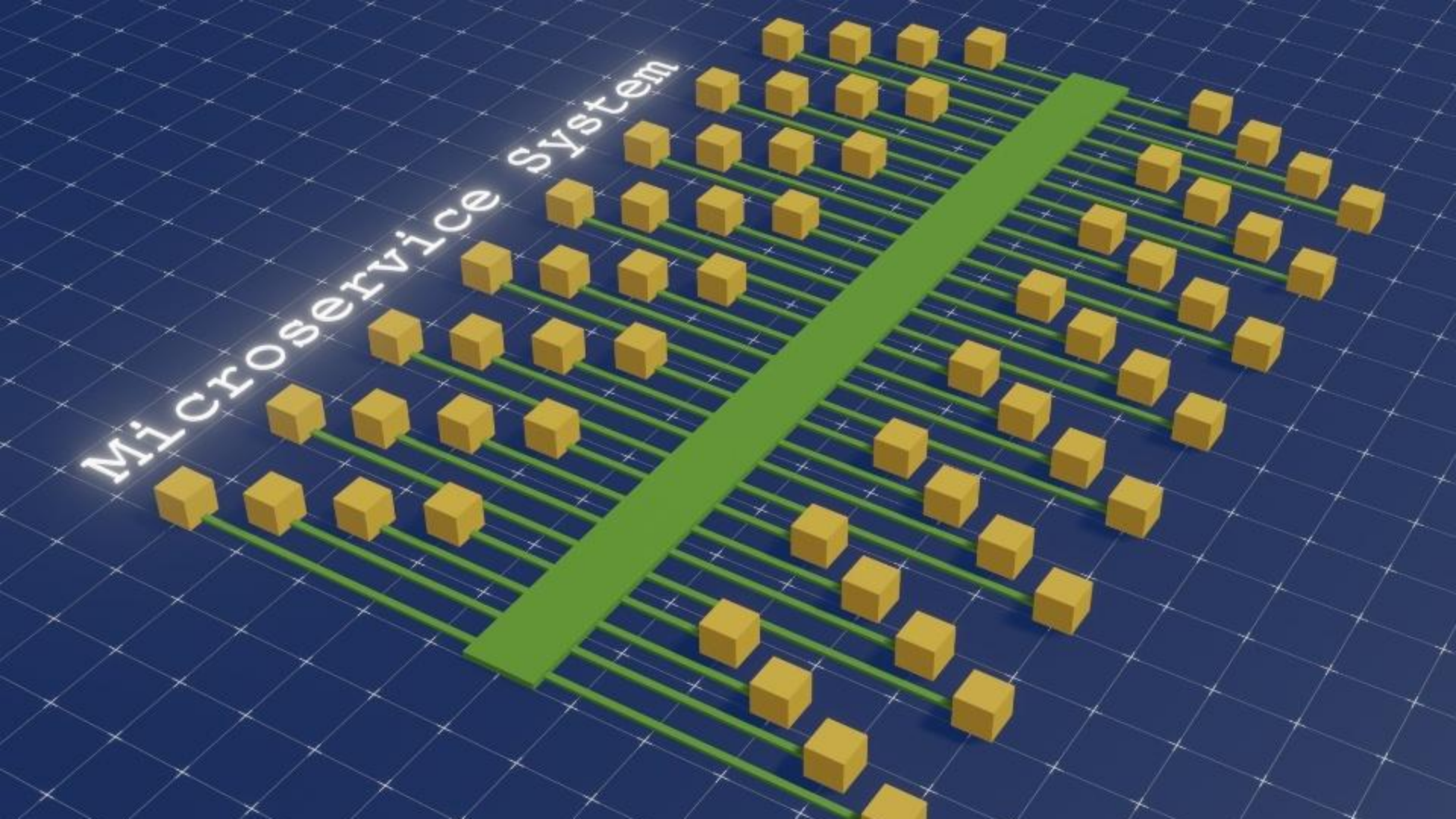
loosley coupled

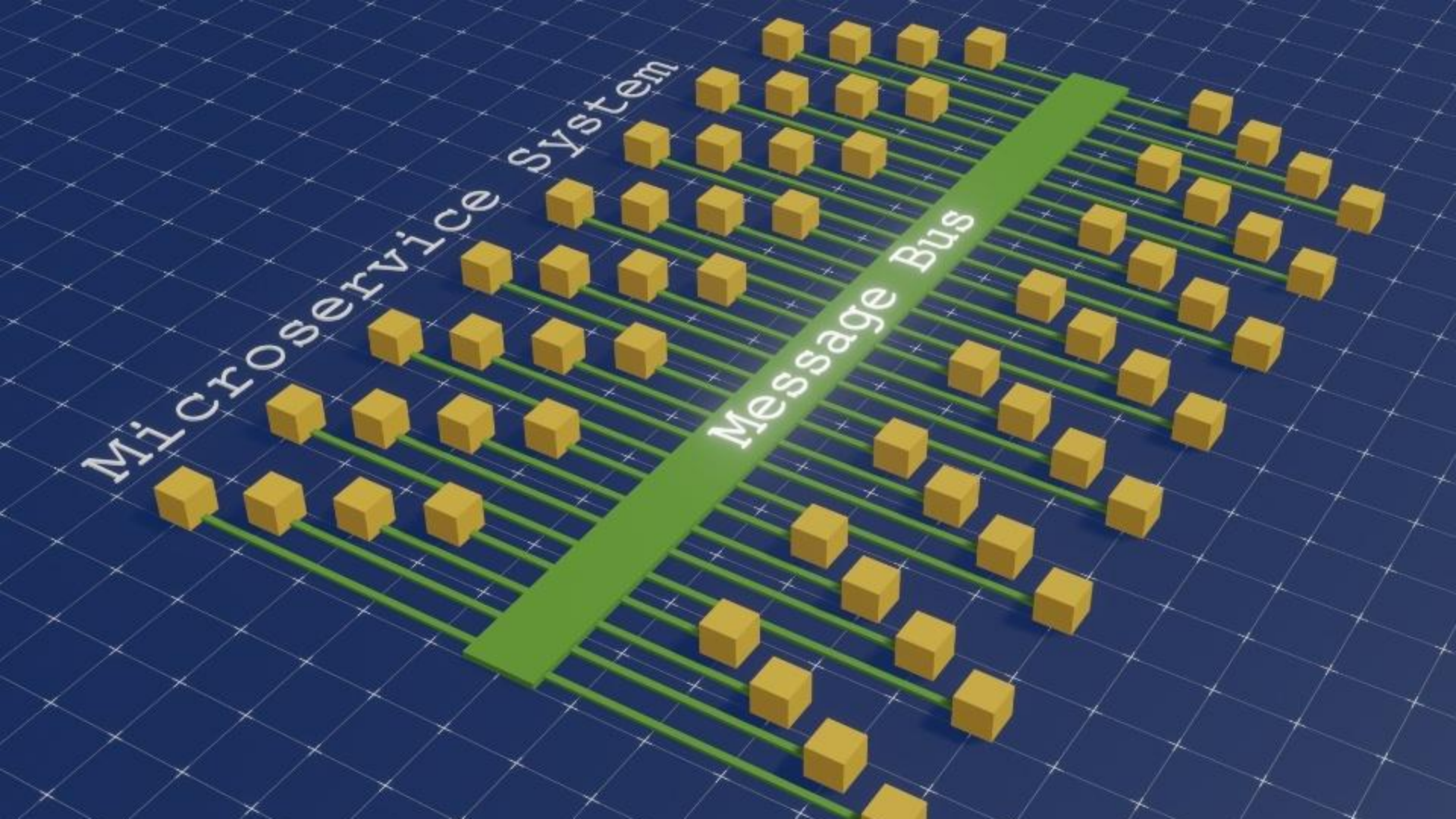

microservice
system

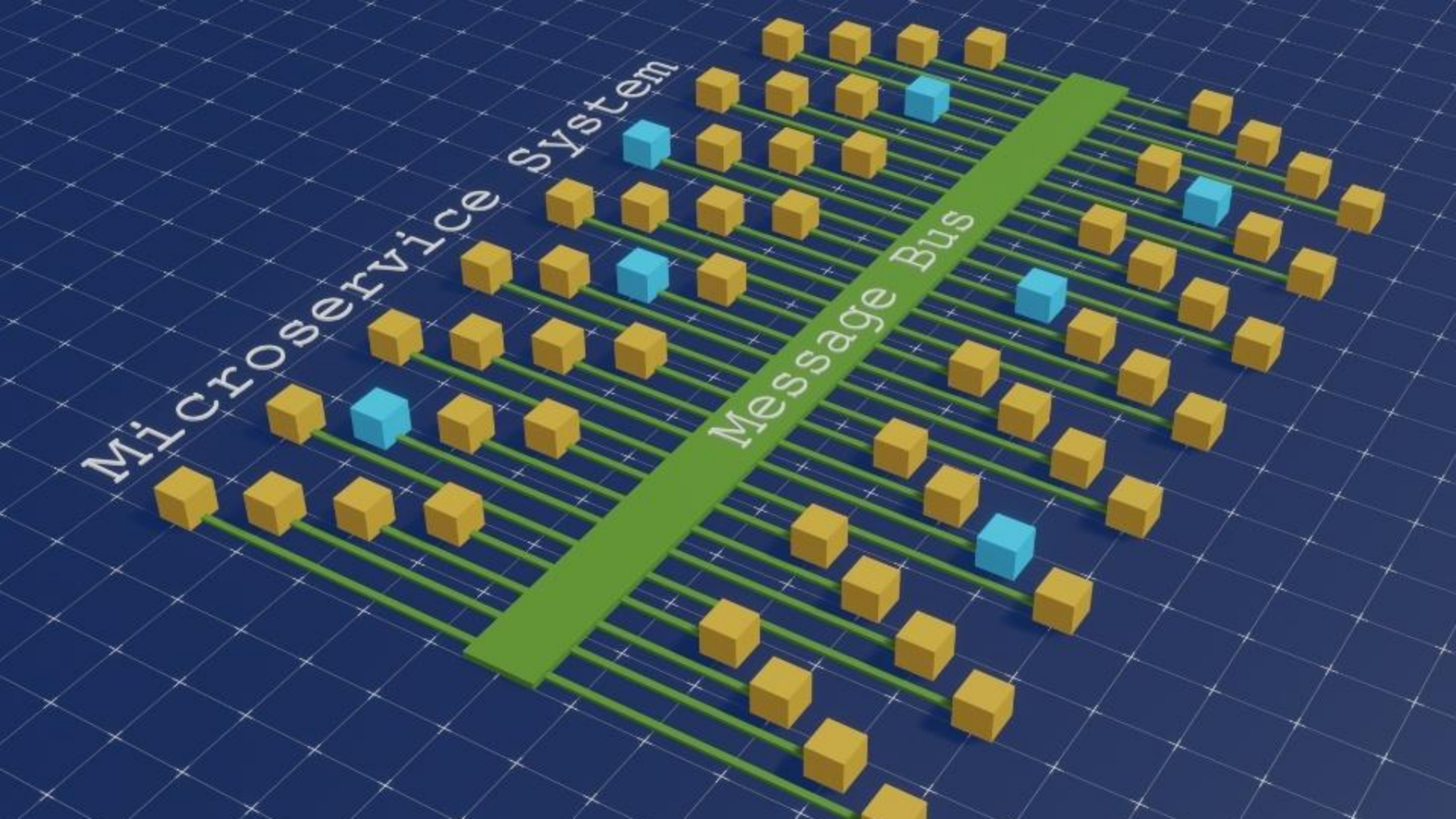


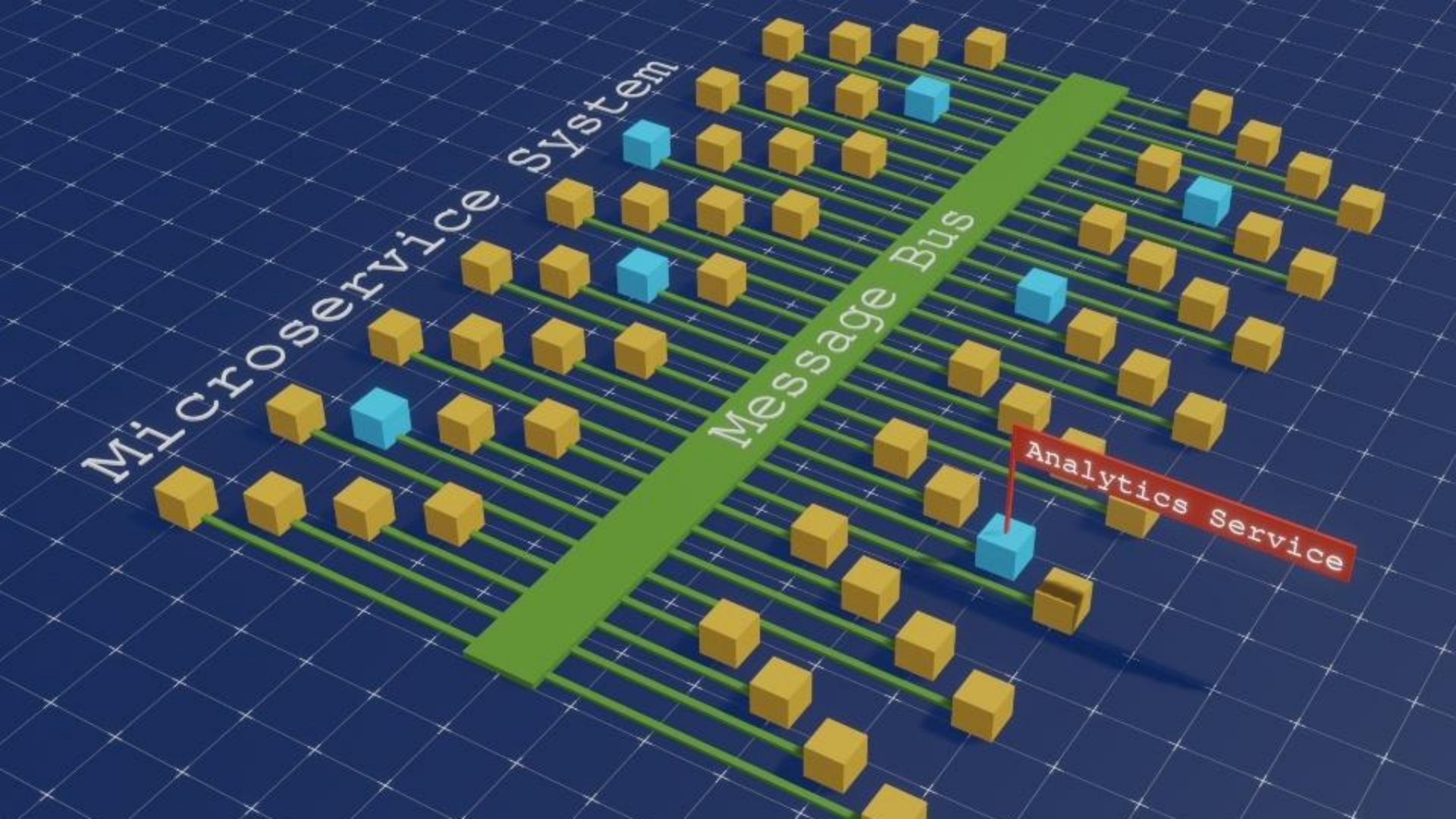


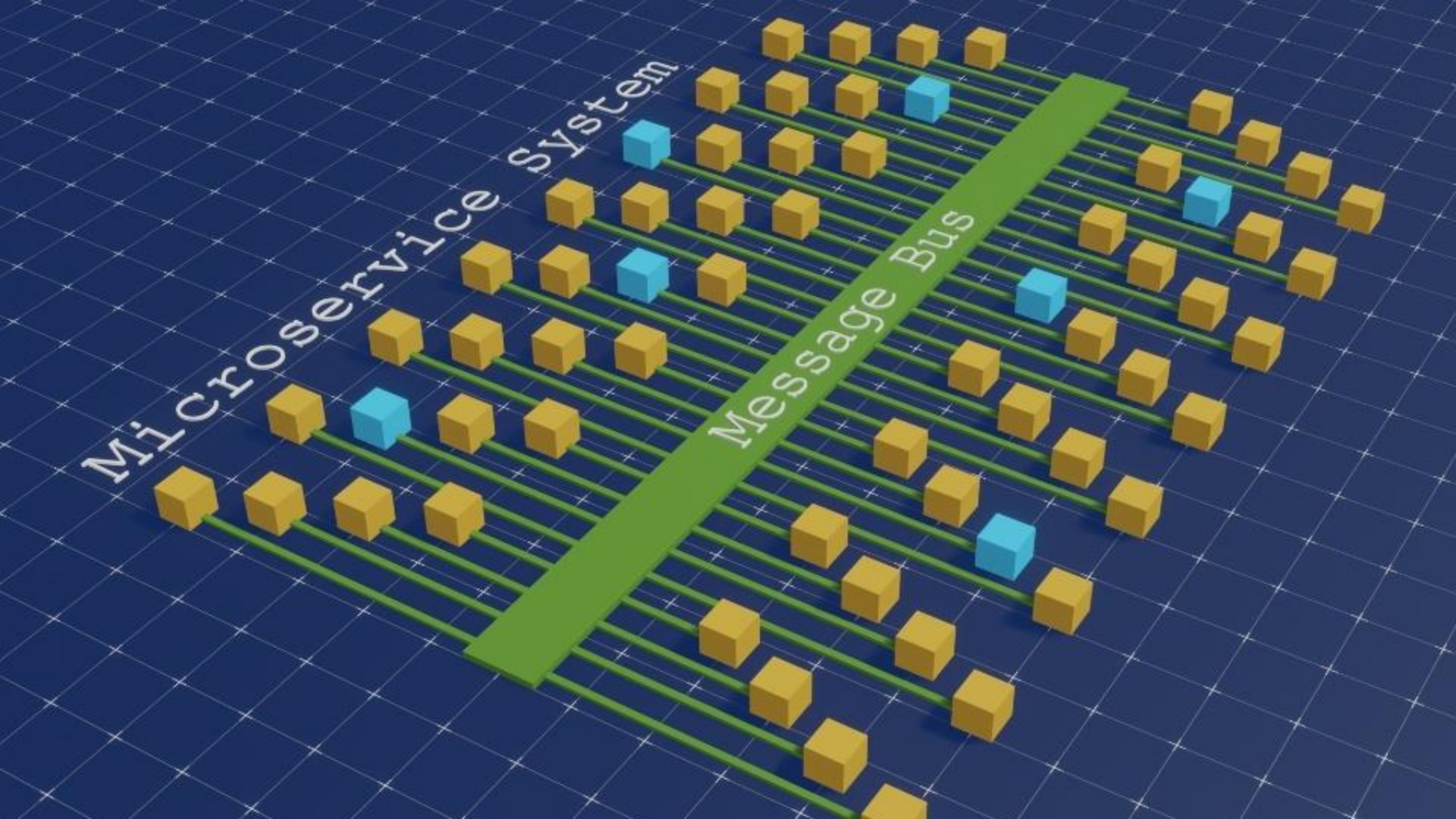


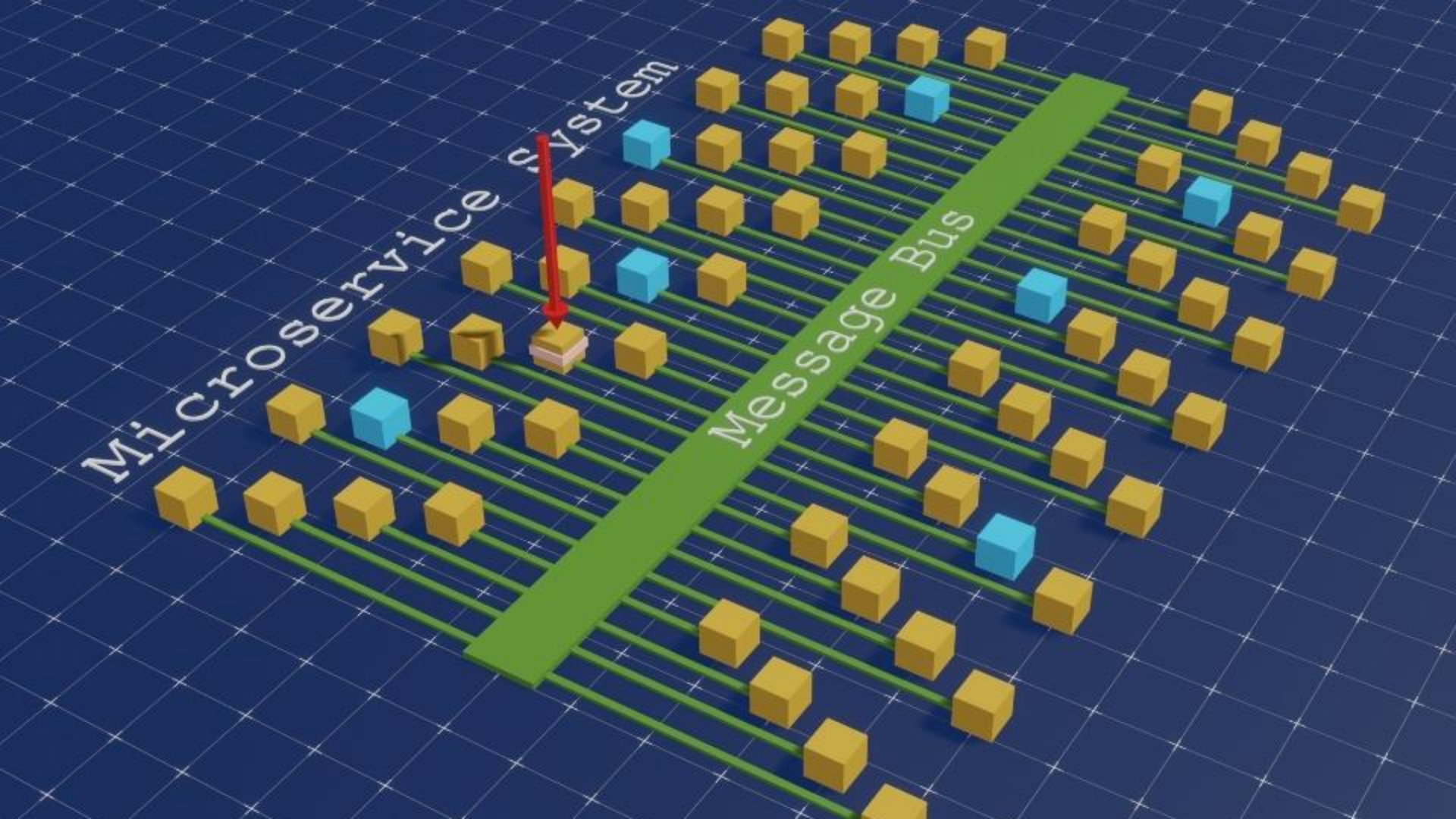

Microservice System

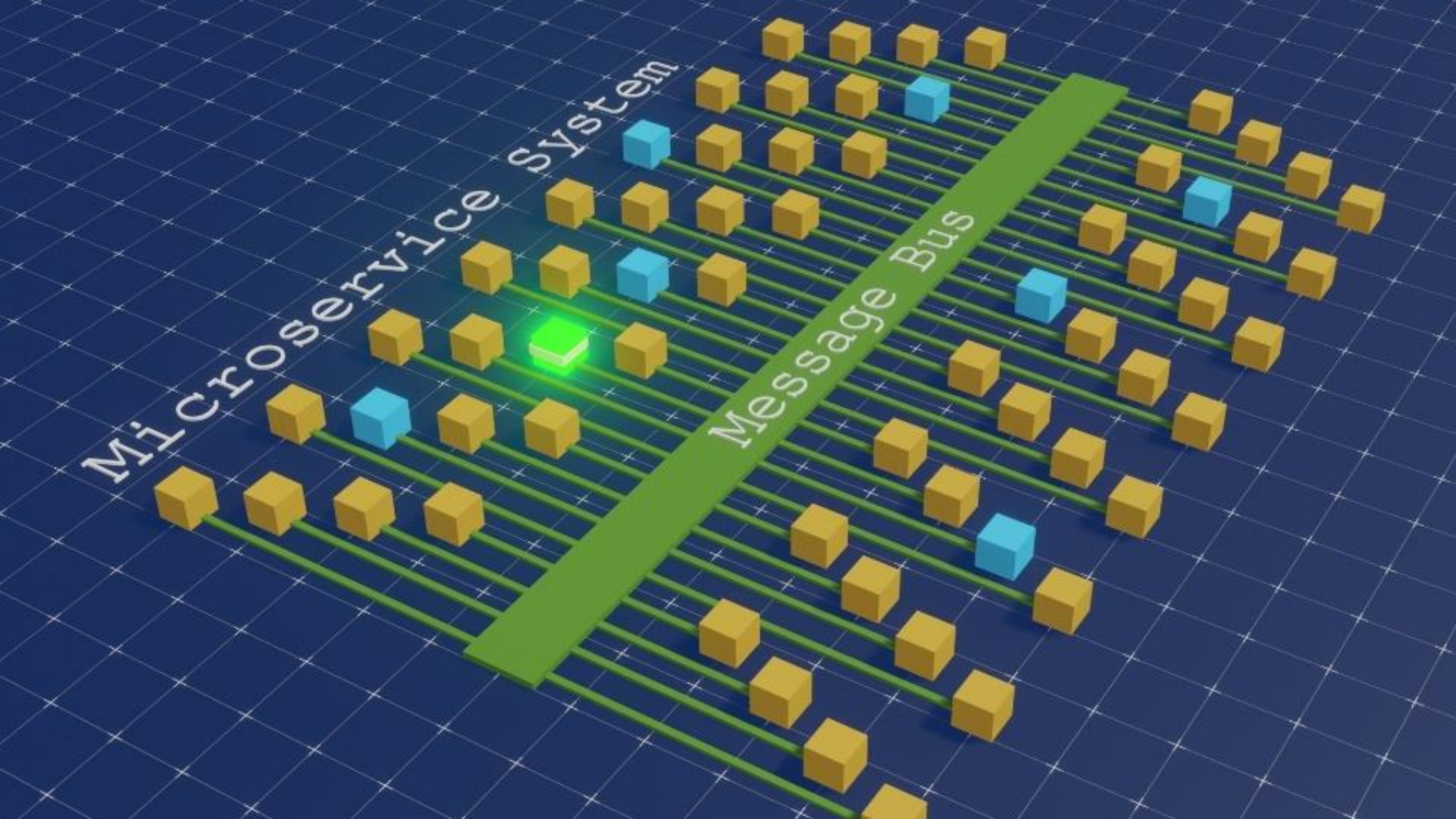

Microservice System
Message Bus

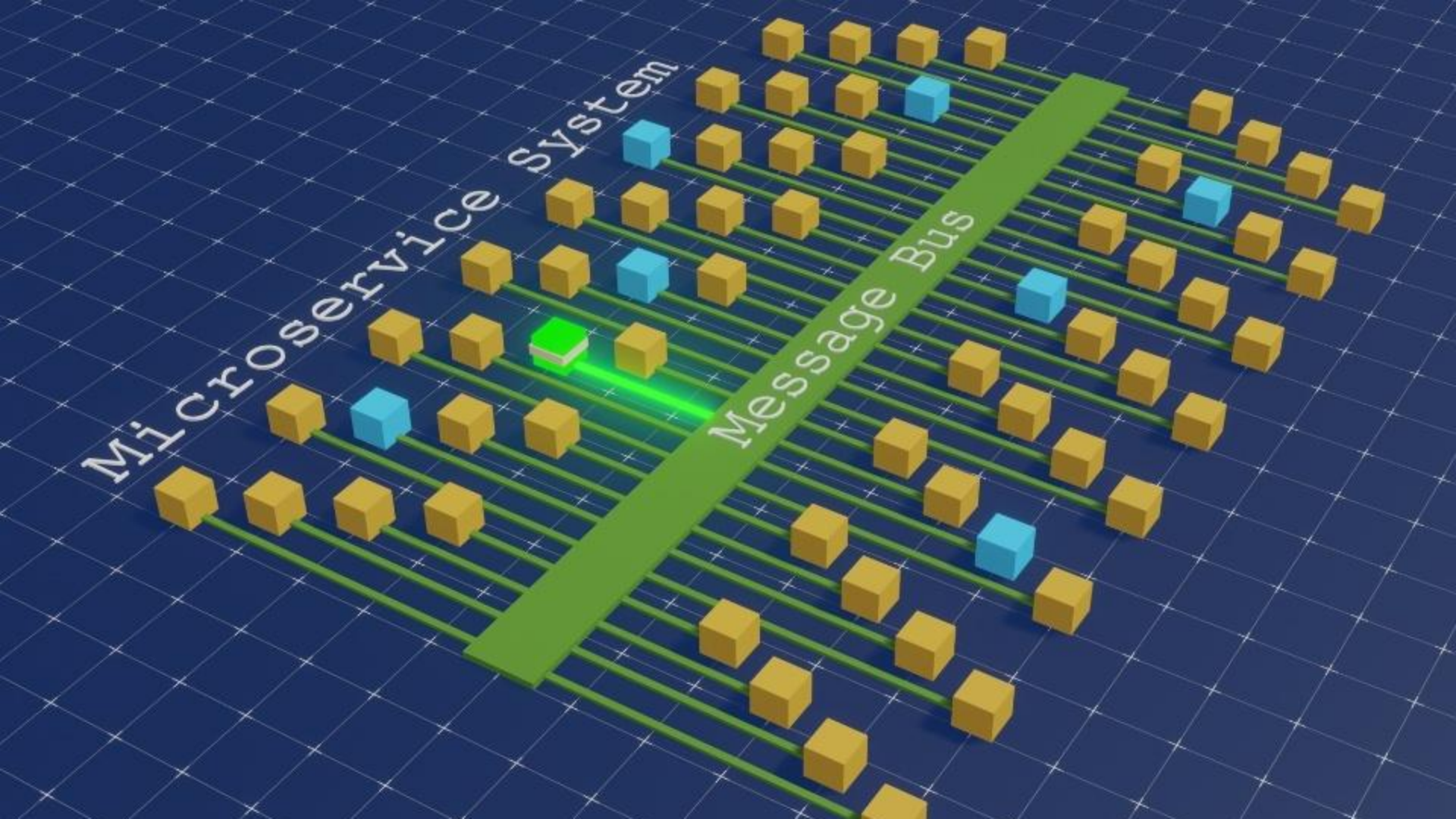

Microservice System
Message Bus

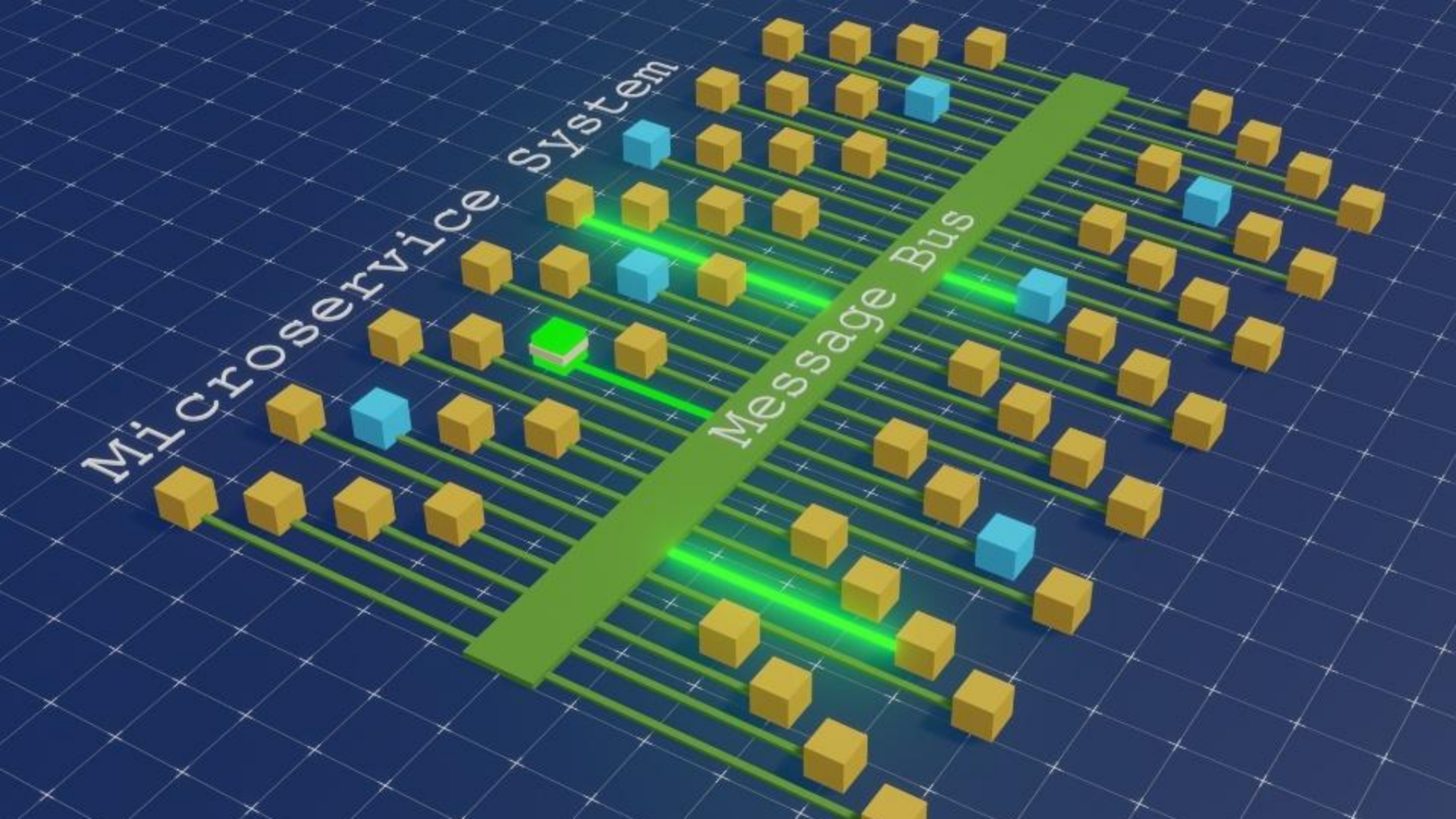

Microservice System
Message Bus
Analytics Service

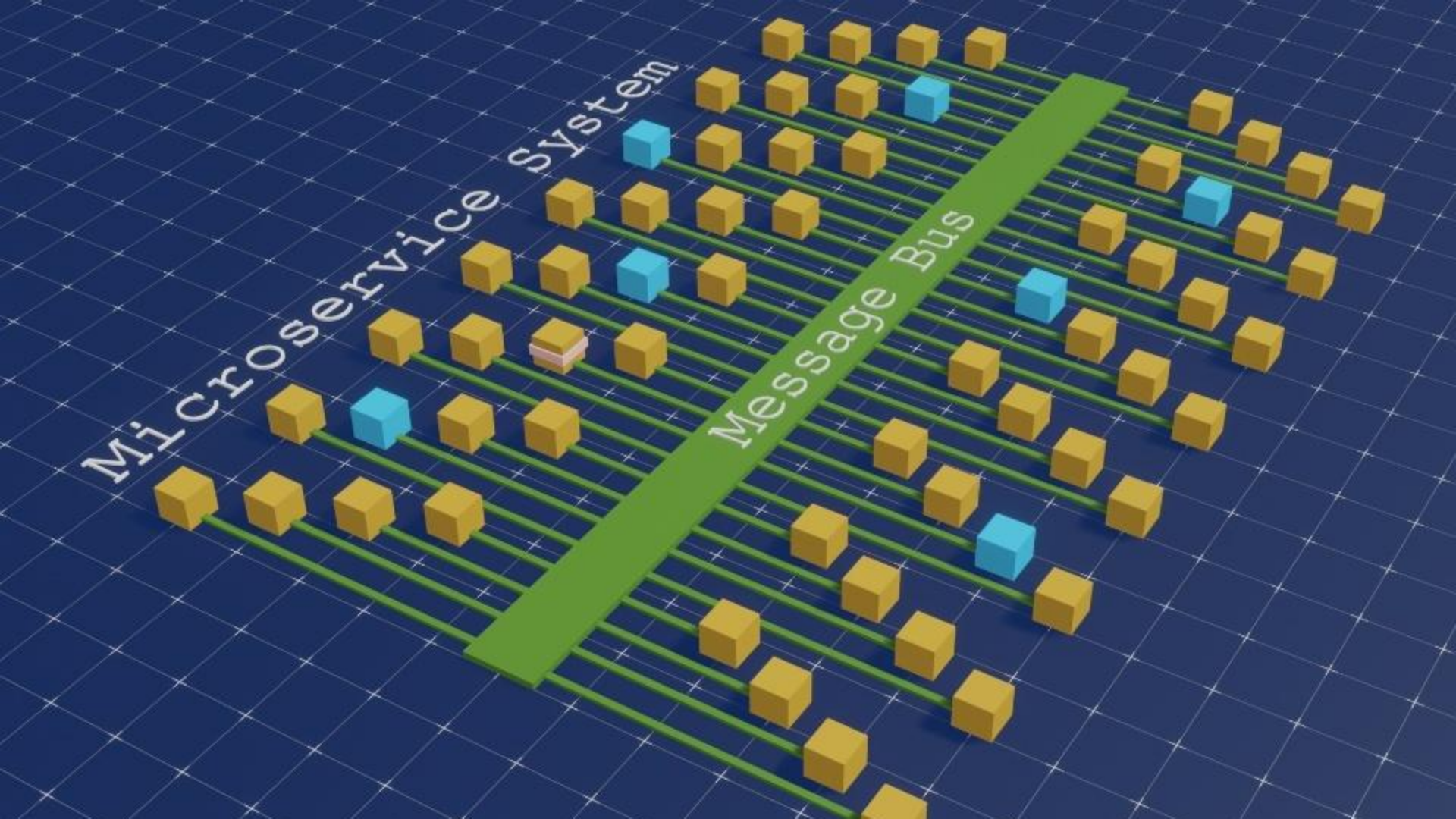

Microservice System
Message Bus

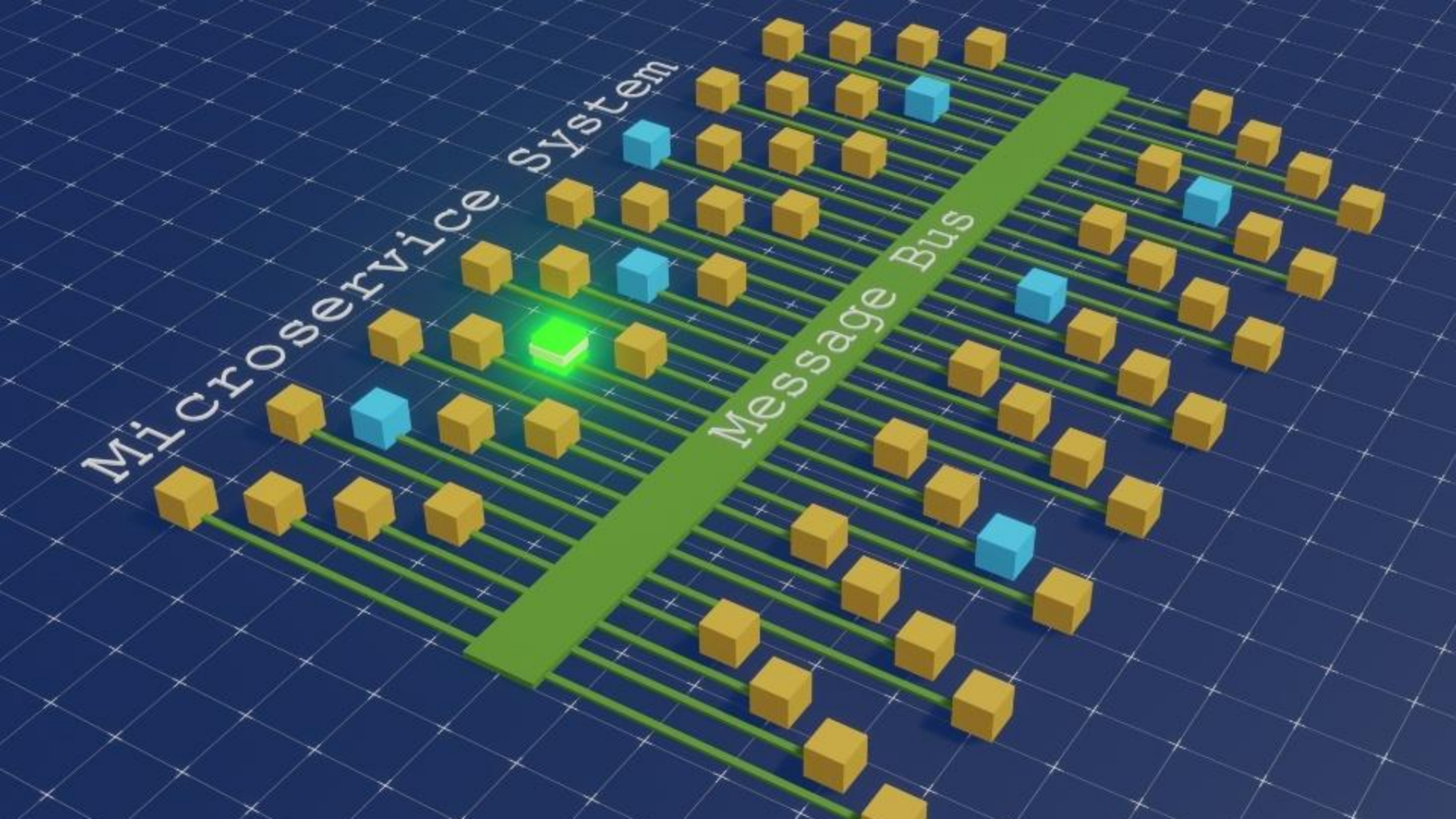

Microservice system
Message Bus

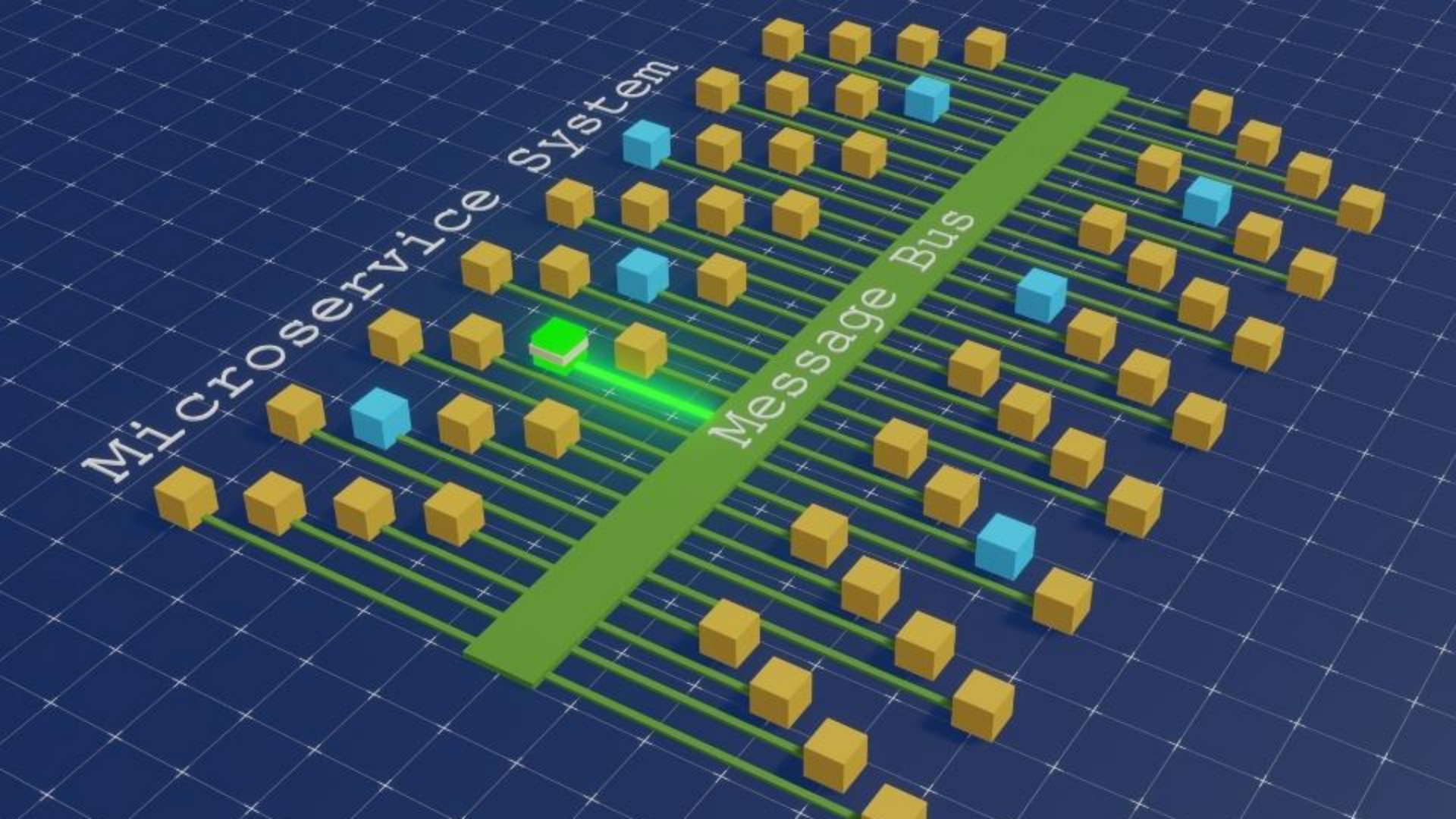

Microservice System
Message Bus

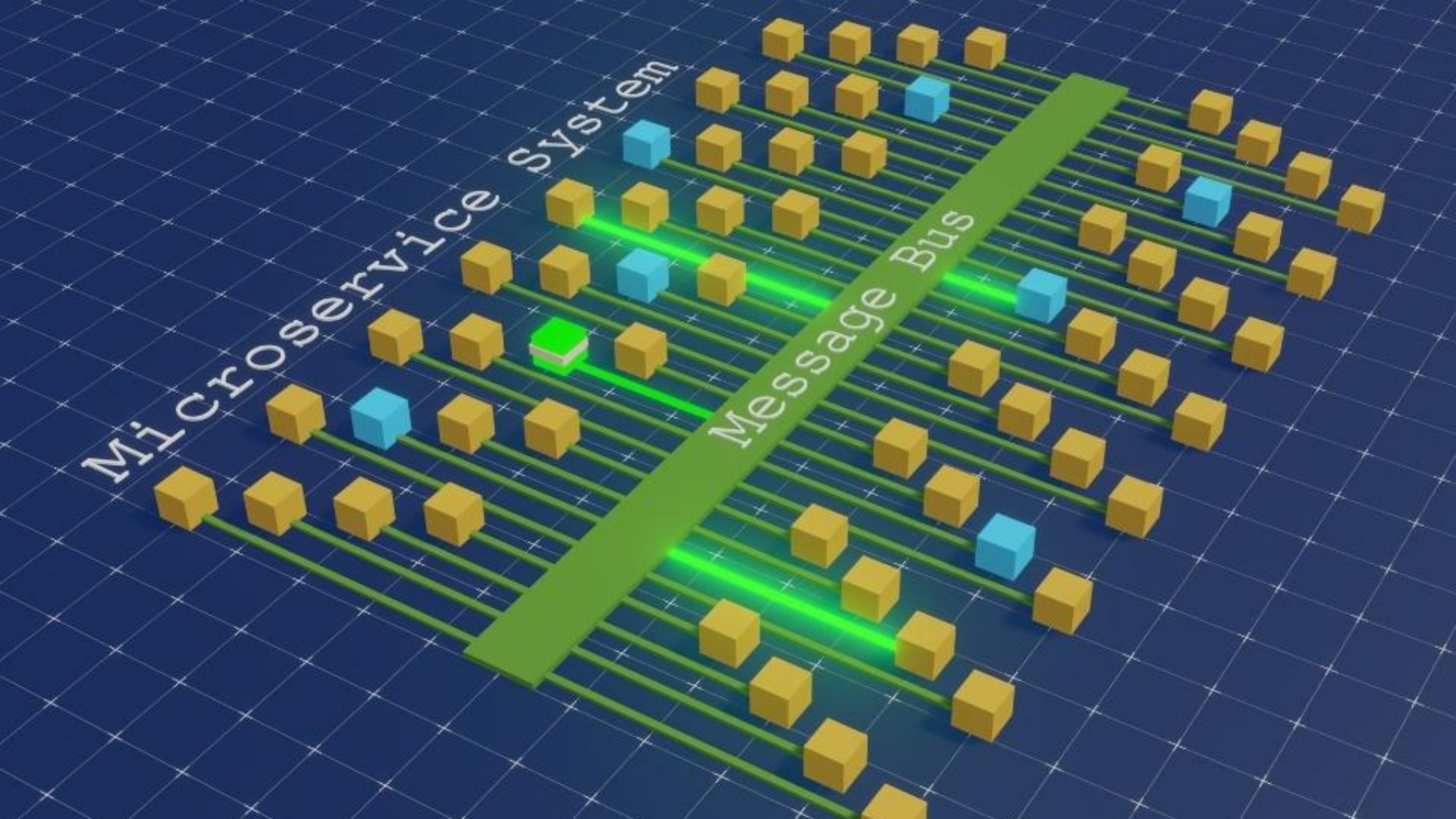

Microservice System
Message Bus

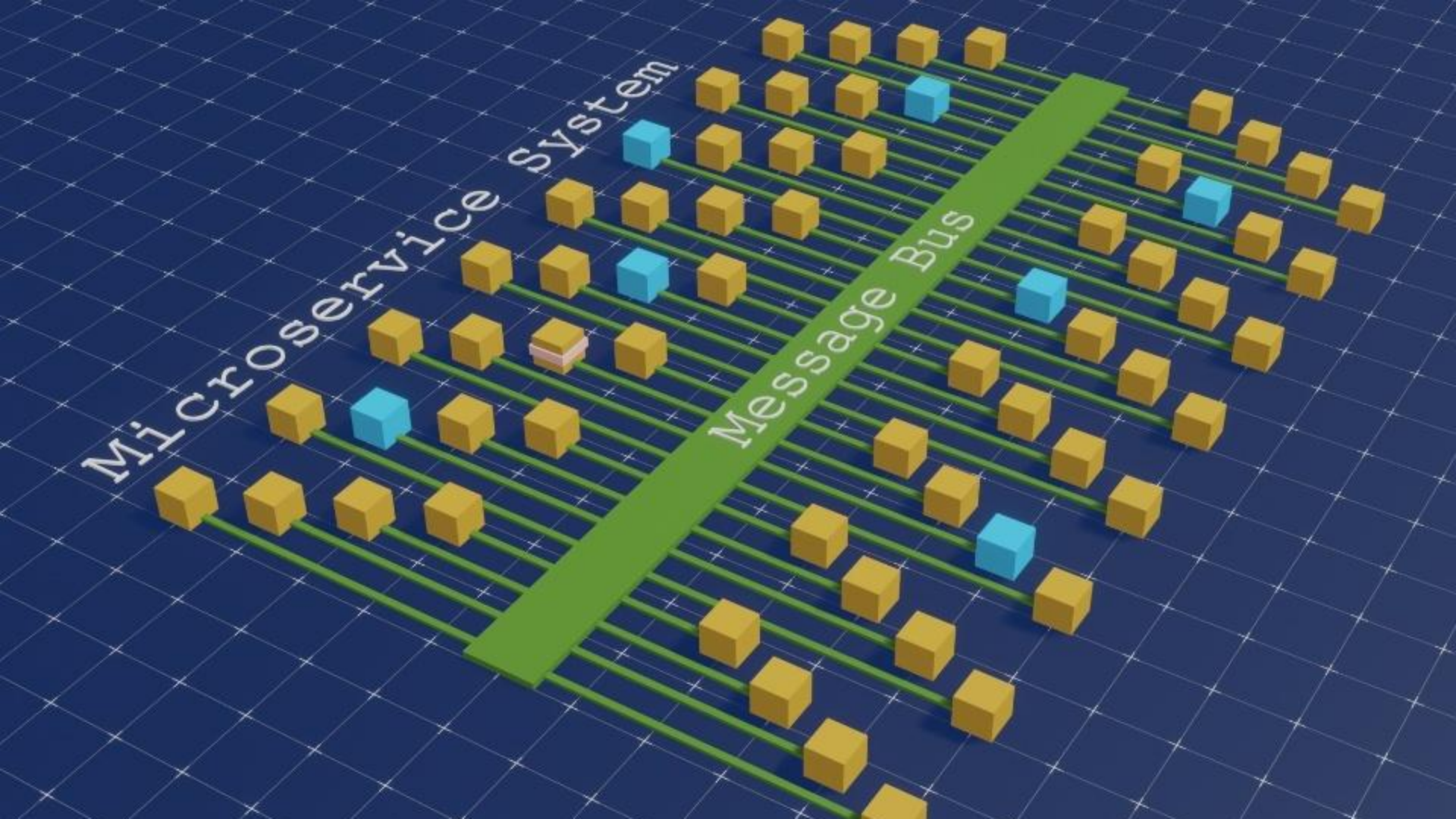

Microservice System
Message Bus

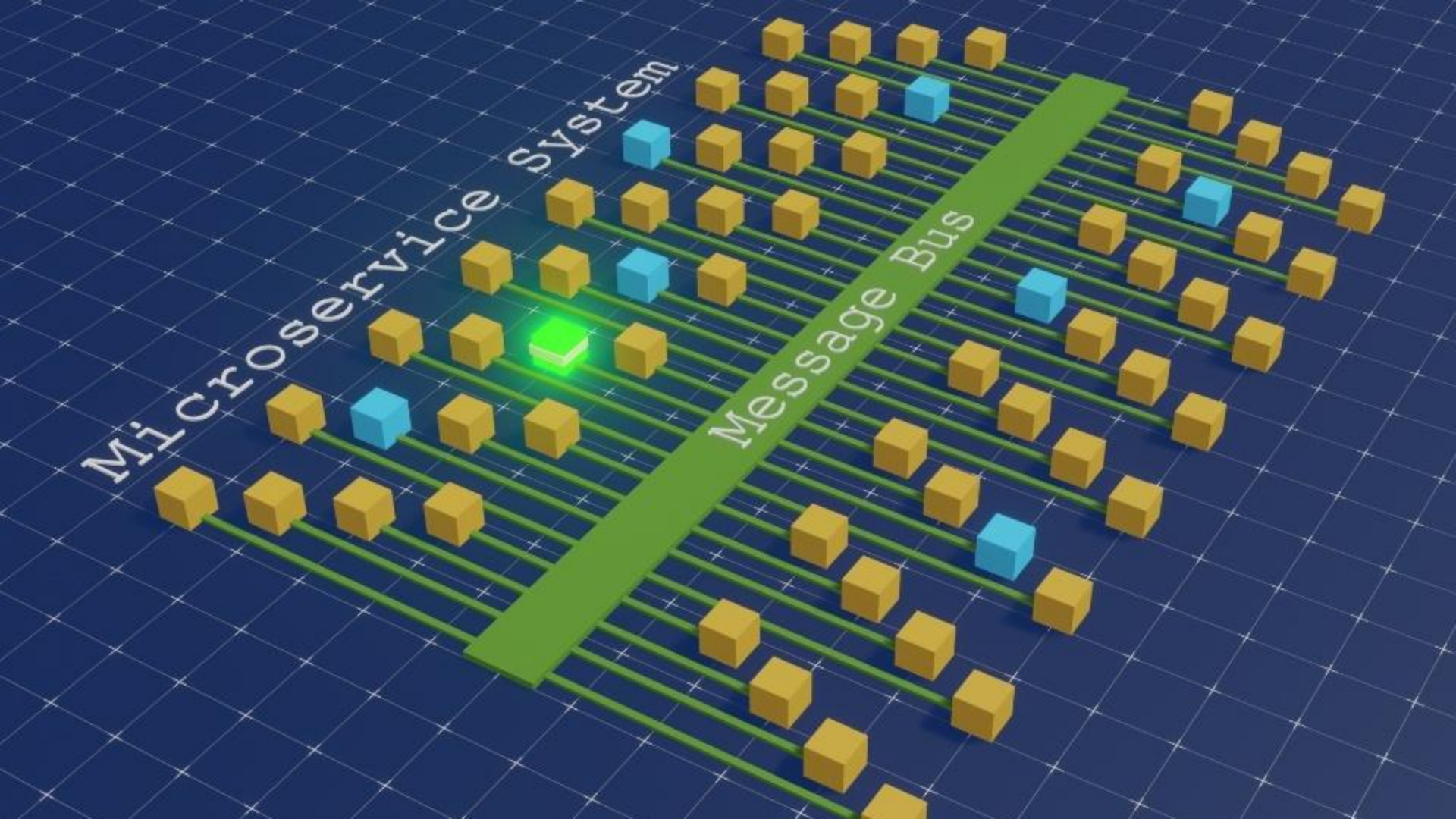

Microservice System
Message Bus

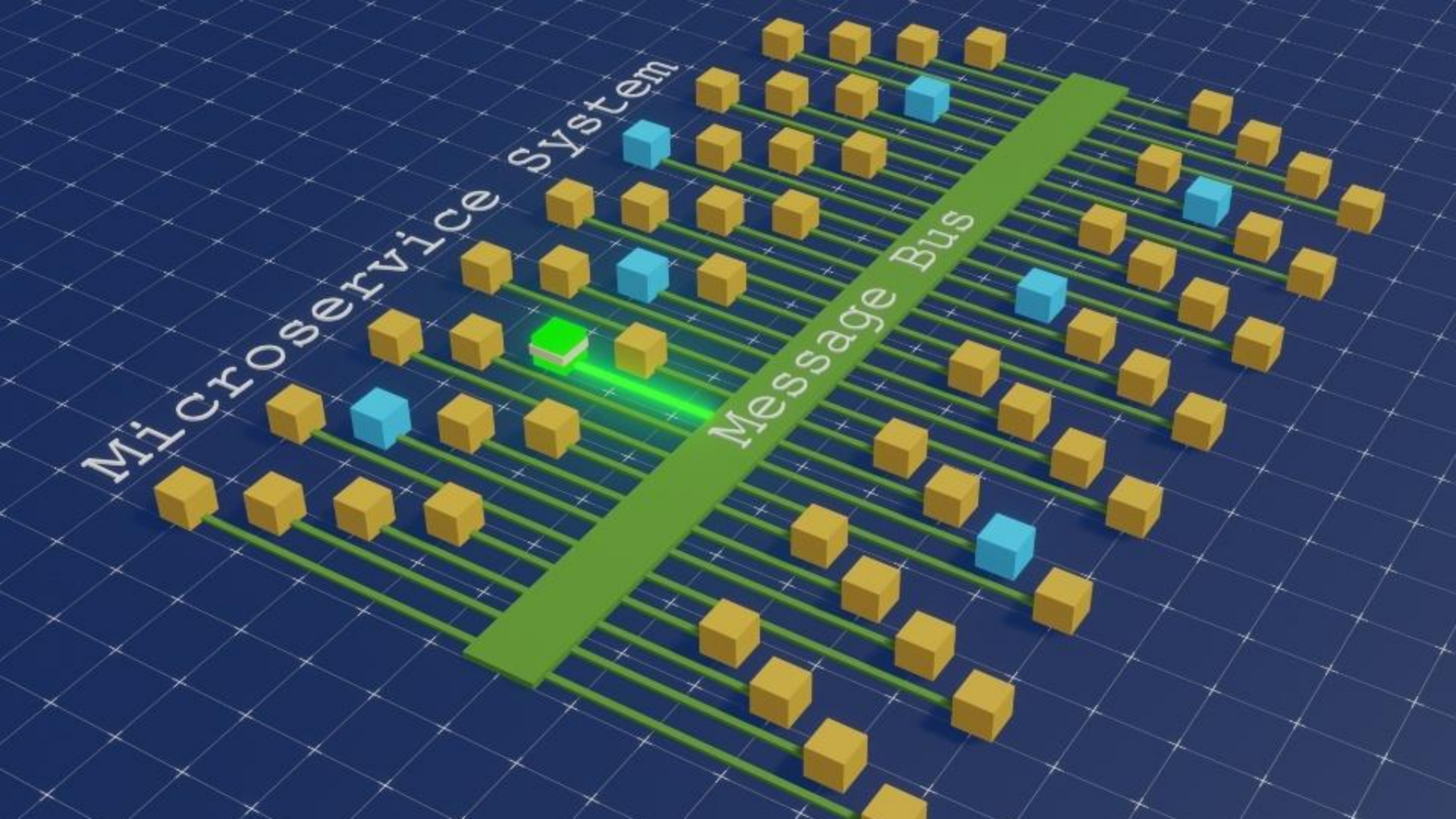

Microservice System
Message Bus

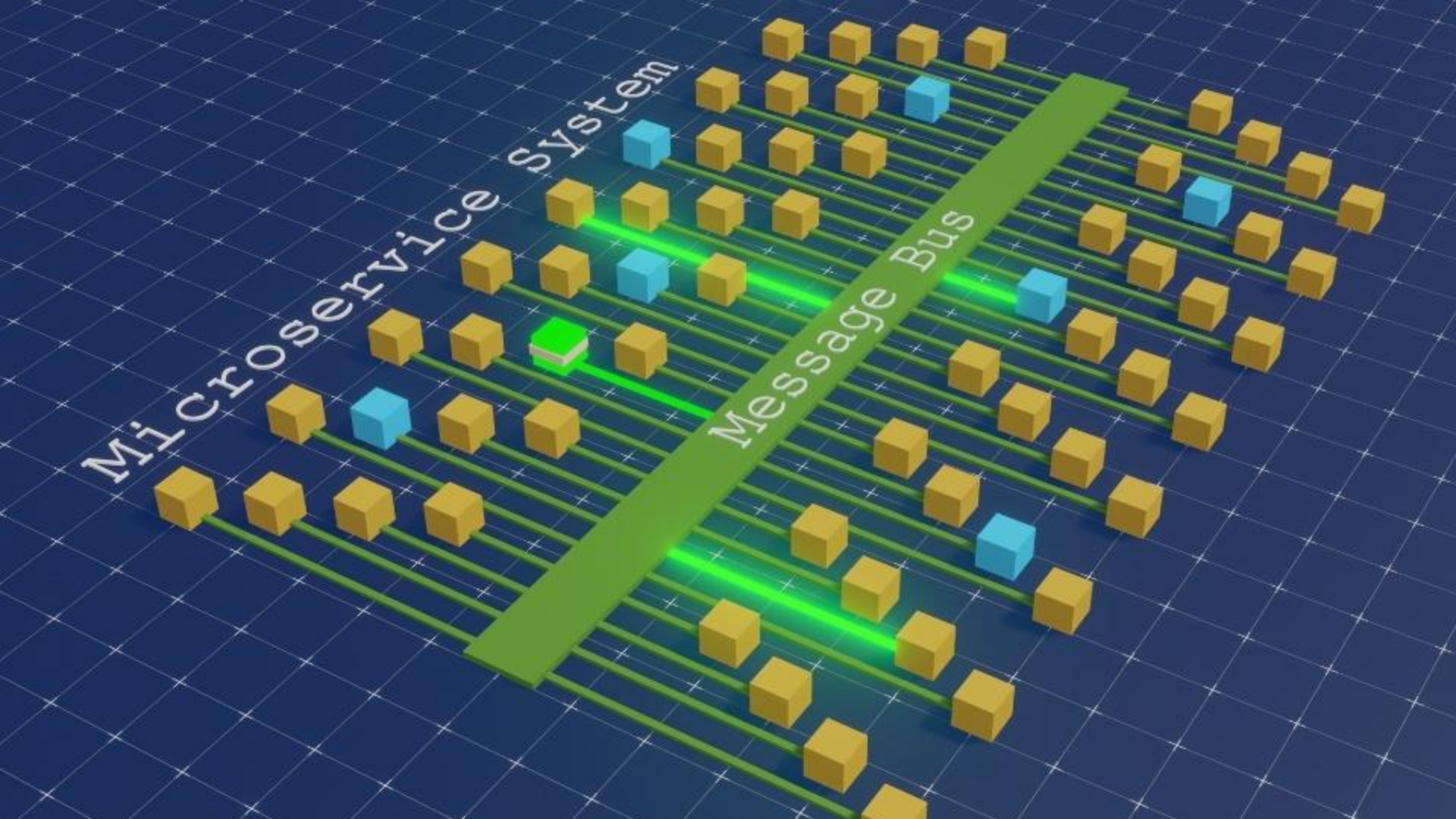

Microservice System
Message Bus

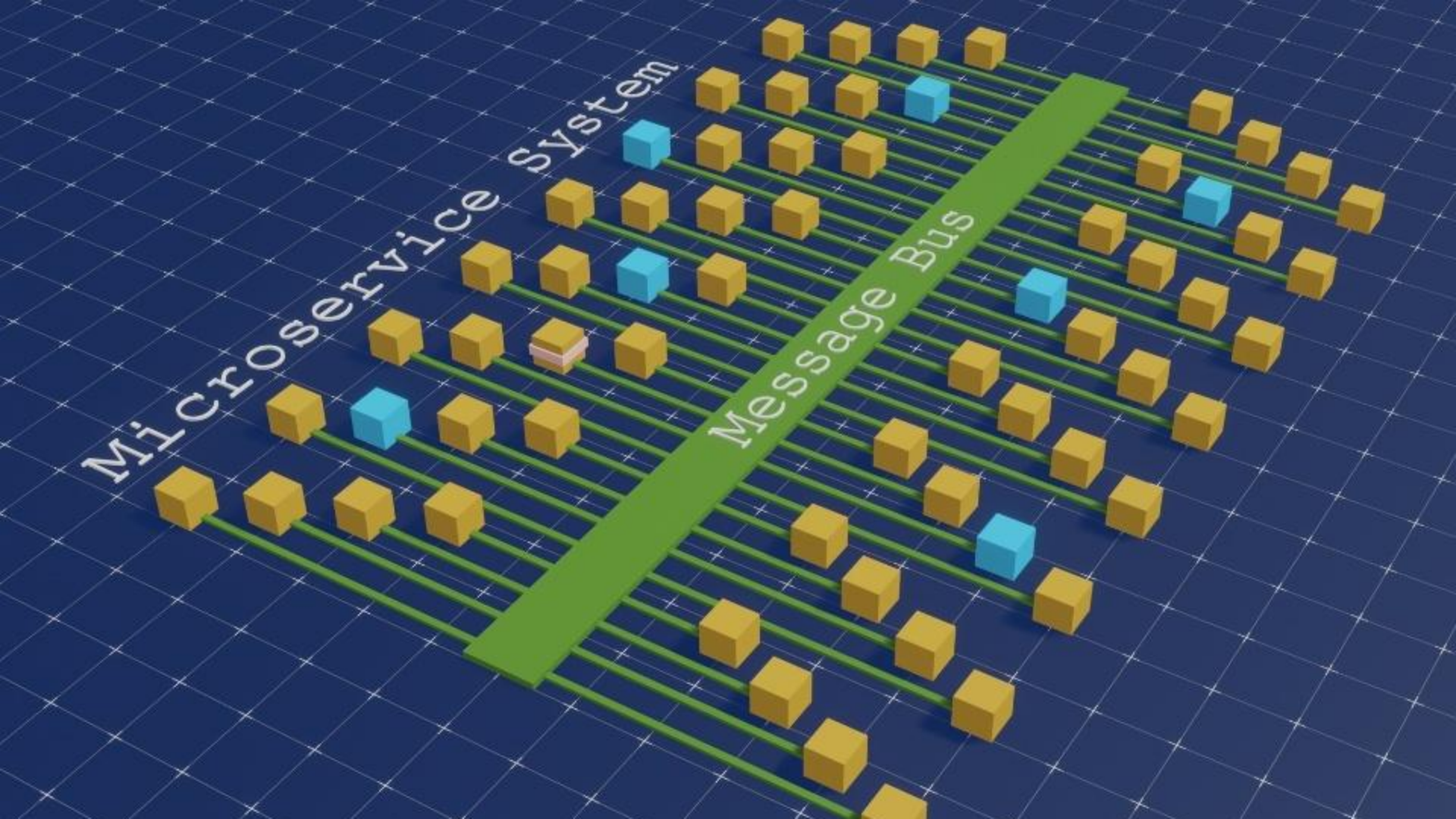

Microservice System
Message Bus

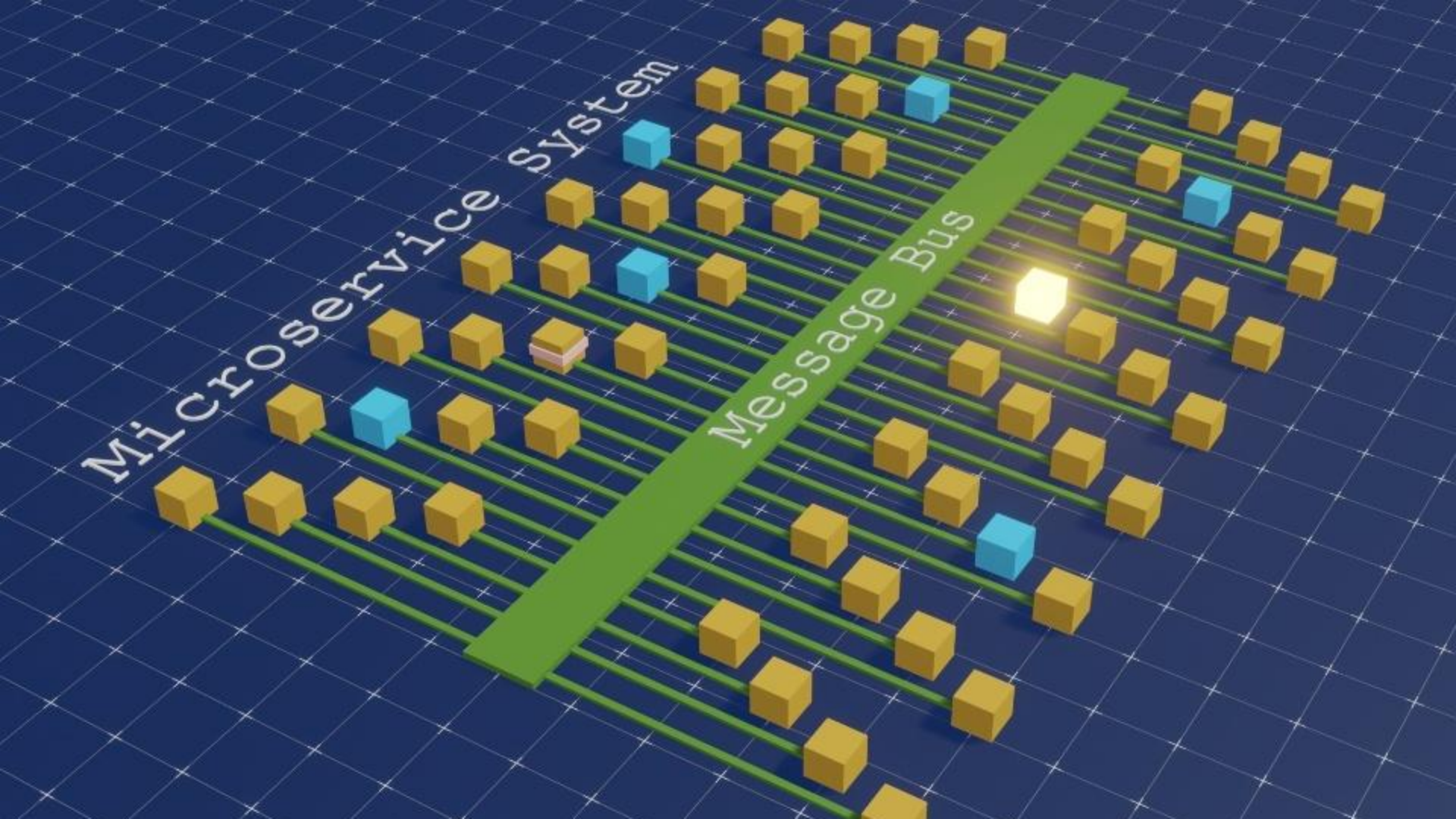

Microservice System
Message Bus

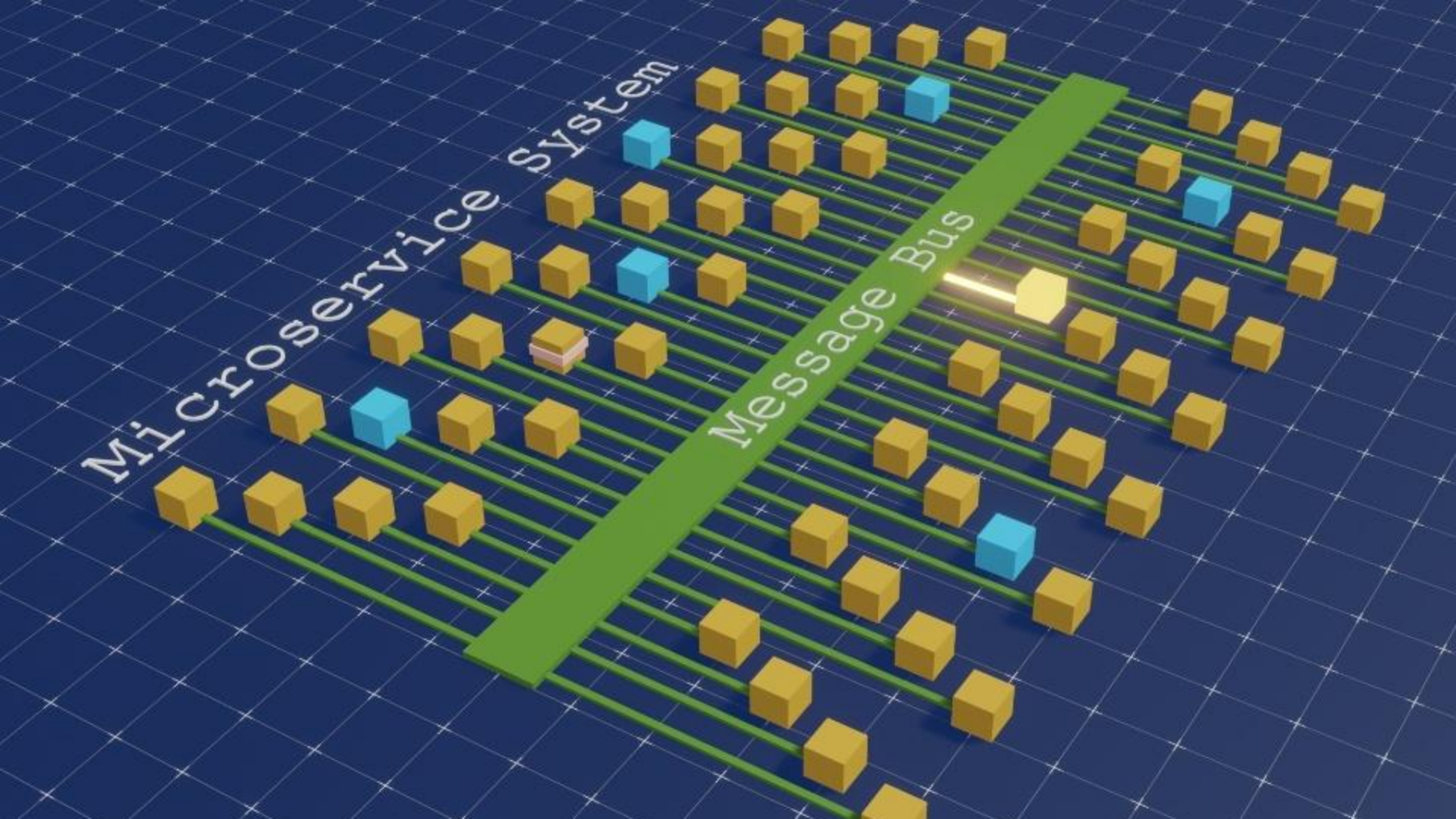

Microservice System
Message Bus

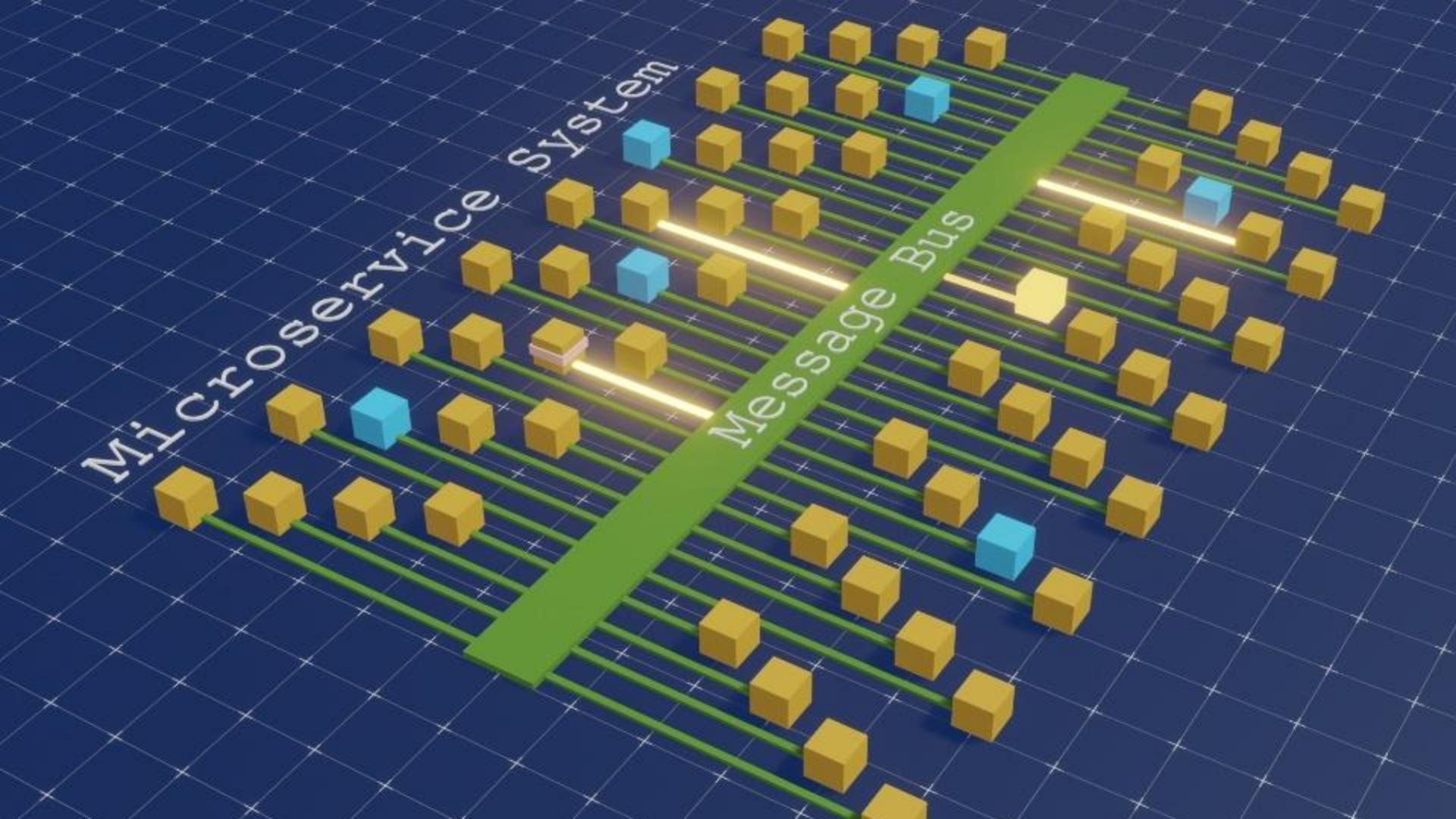

Microservice System
Message Bus

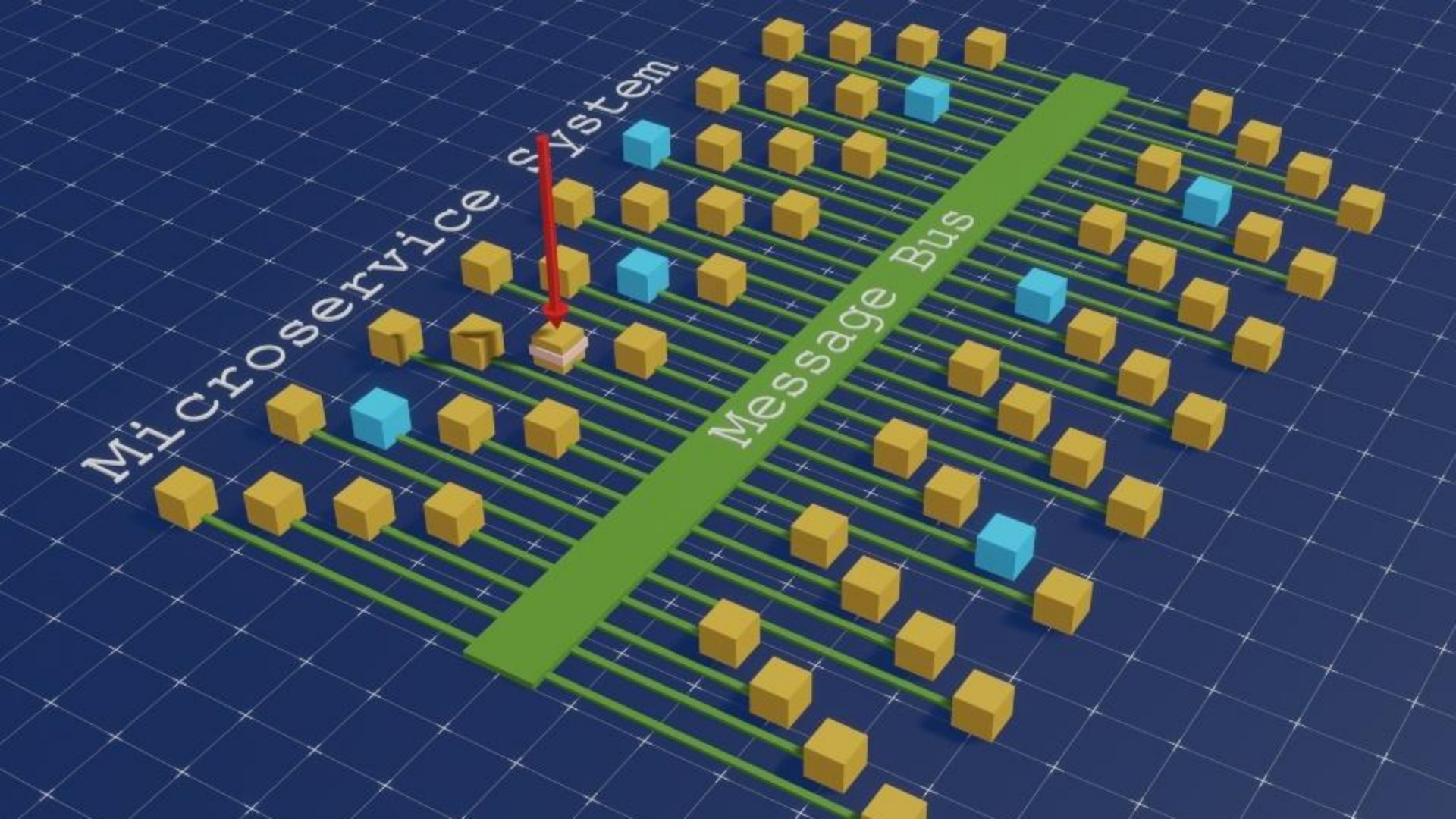

Microservice System
Message Bus

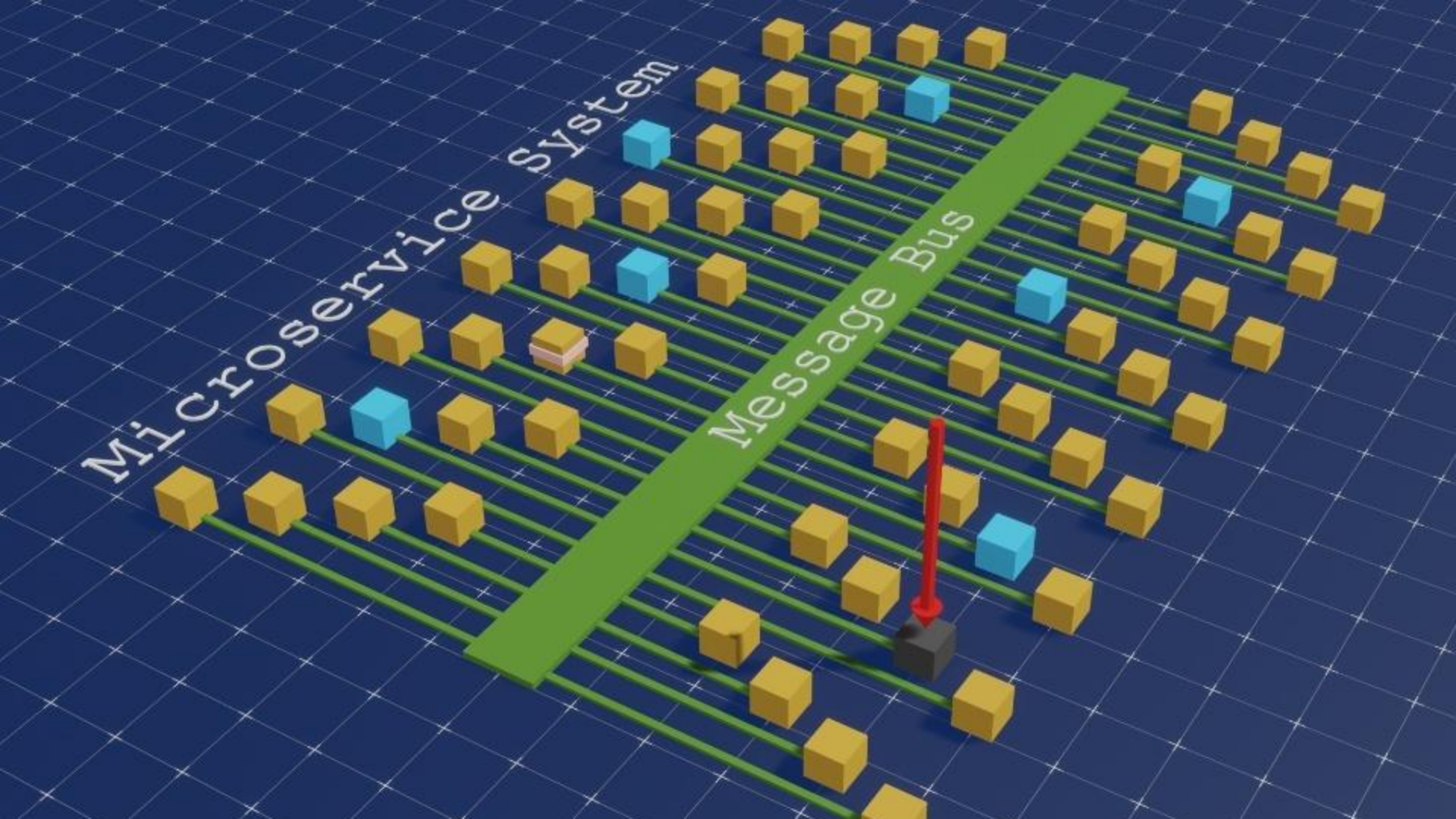

Microservice System
Message Bus

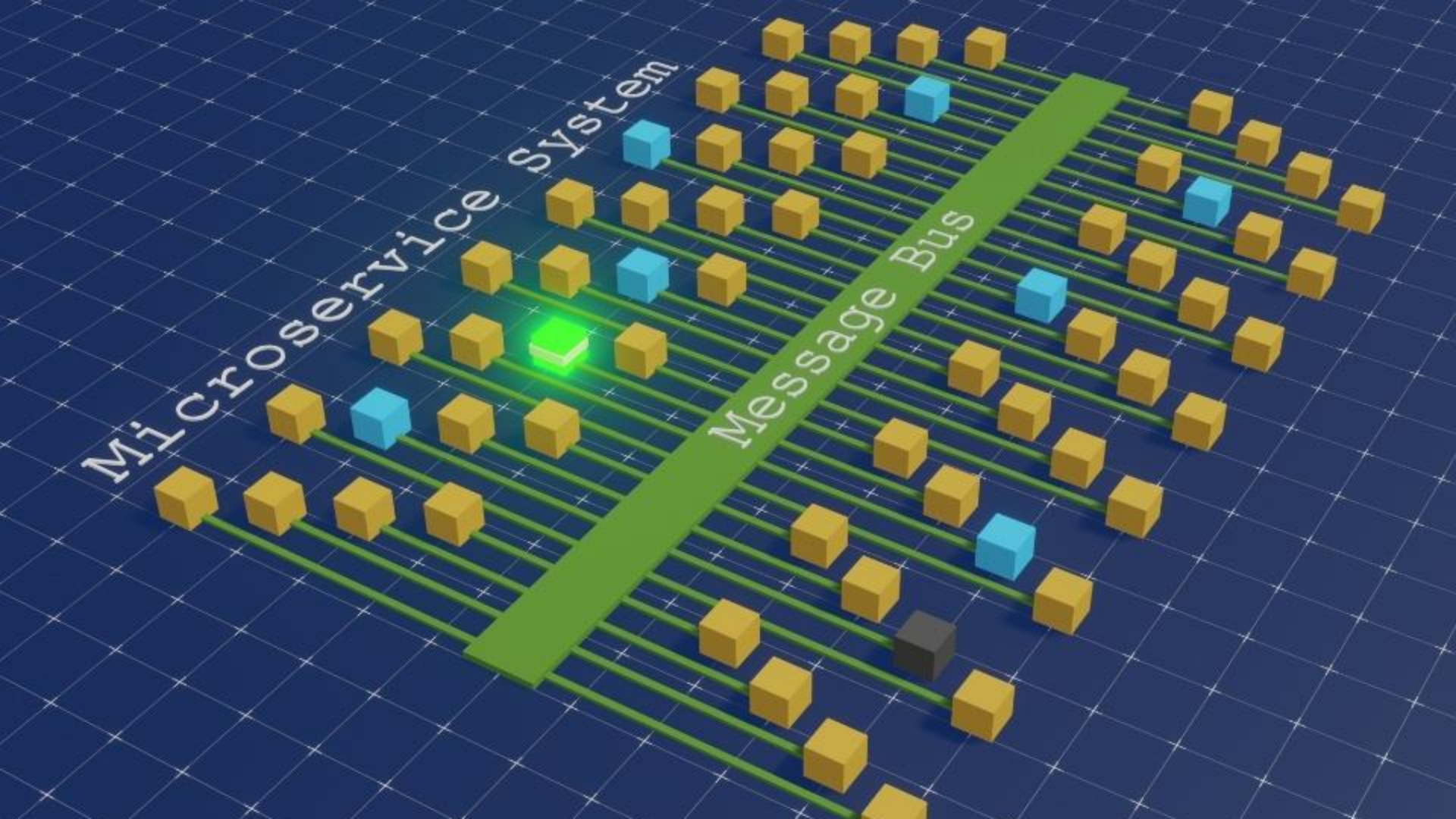

Microservice System
Message Bus

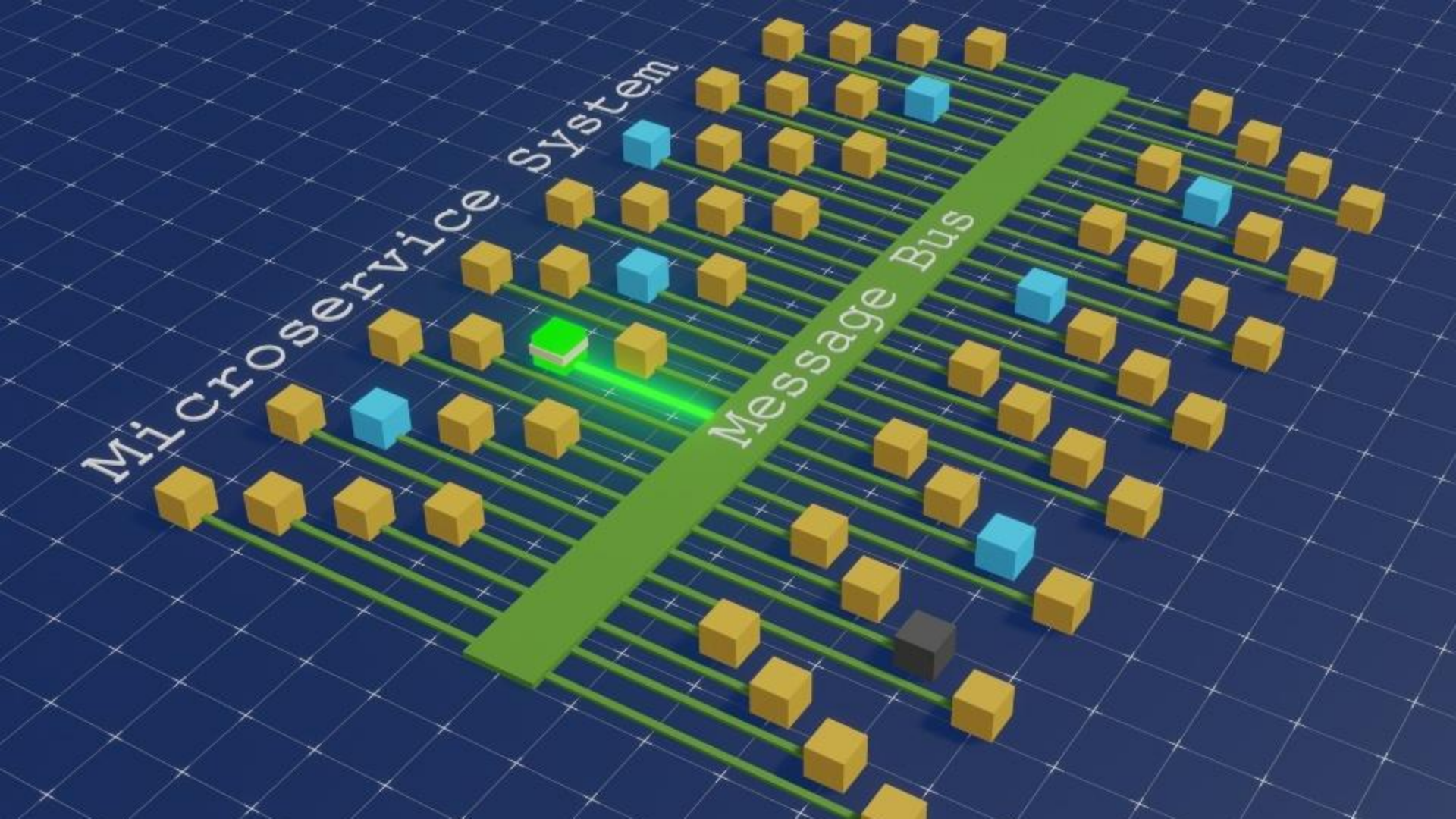

Microservice System
Message Bus

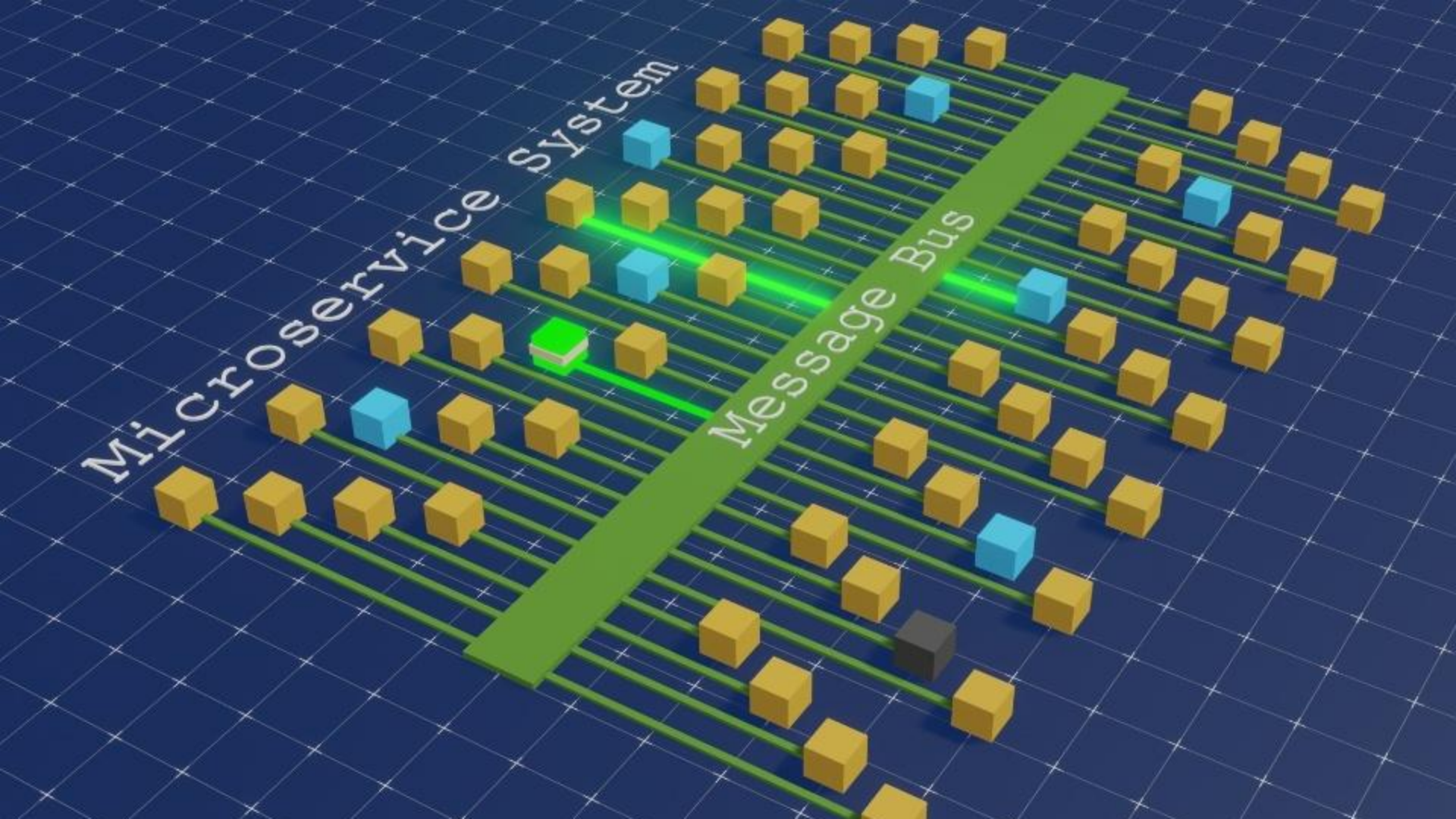

Microservice System
Message Bus

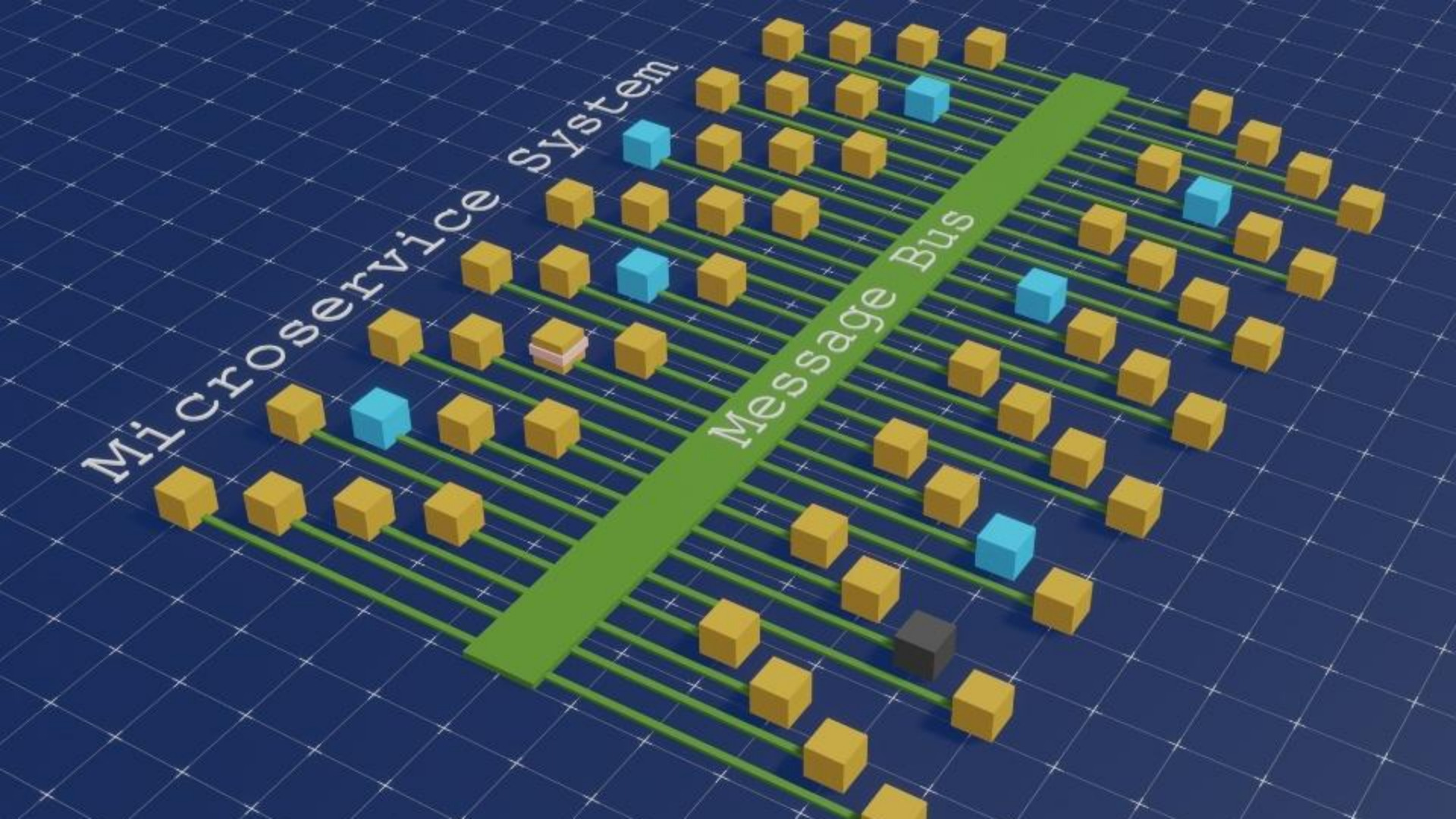

Microservice system
Message Bus

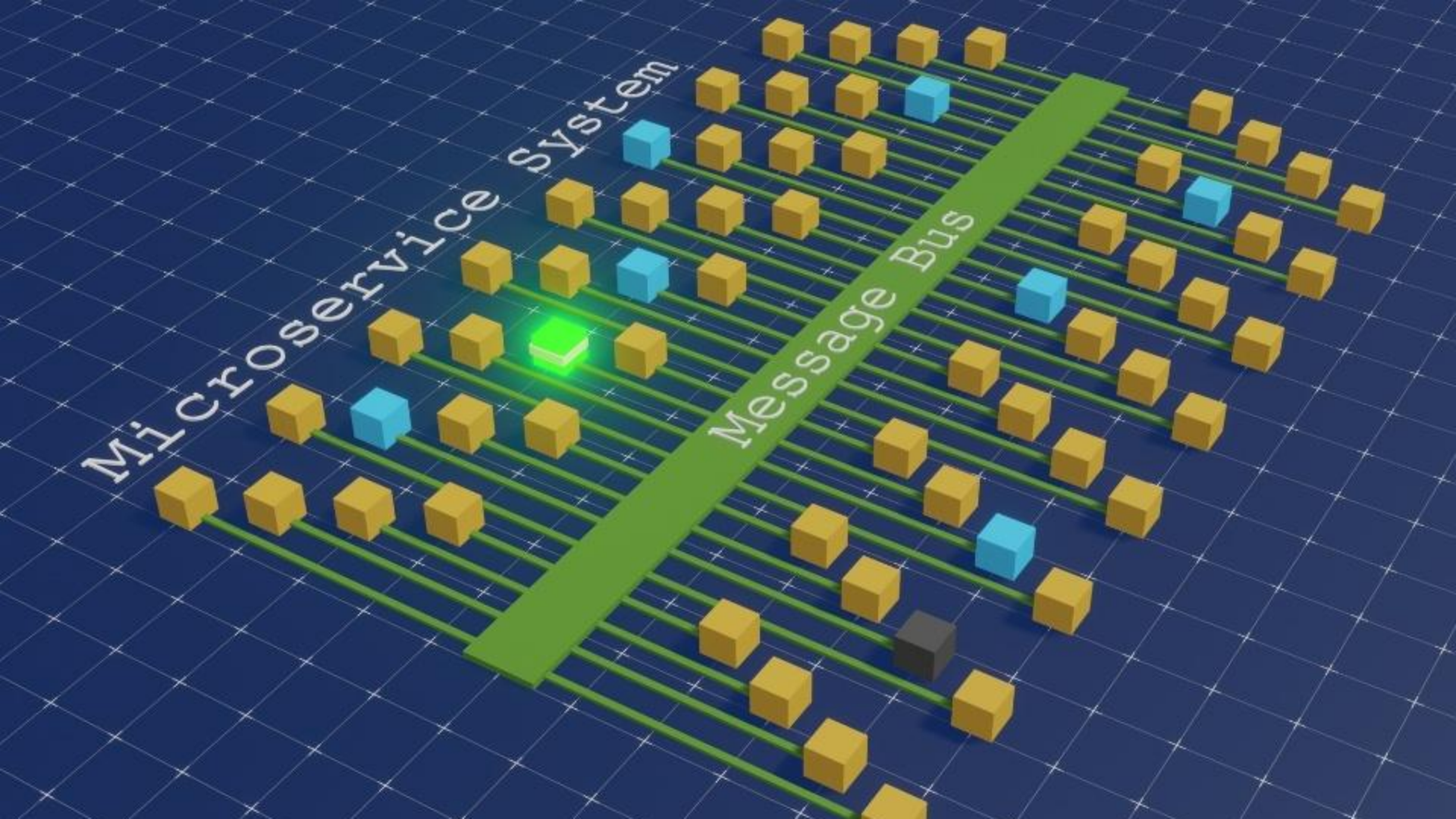

Microservice System
Message Bus

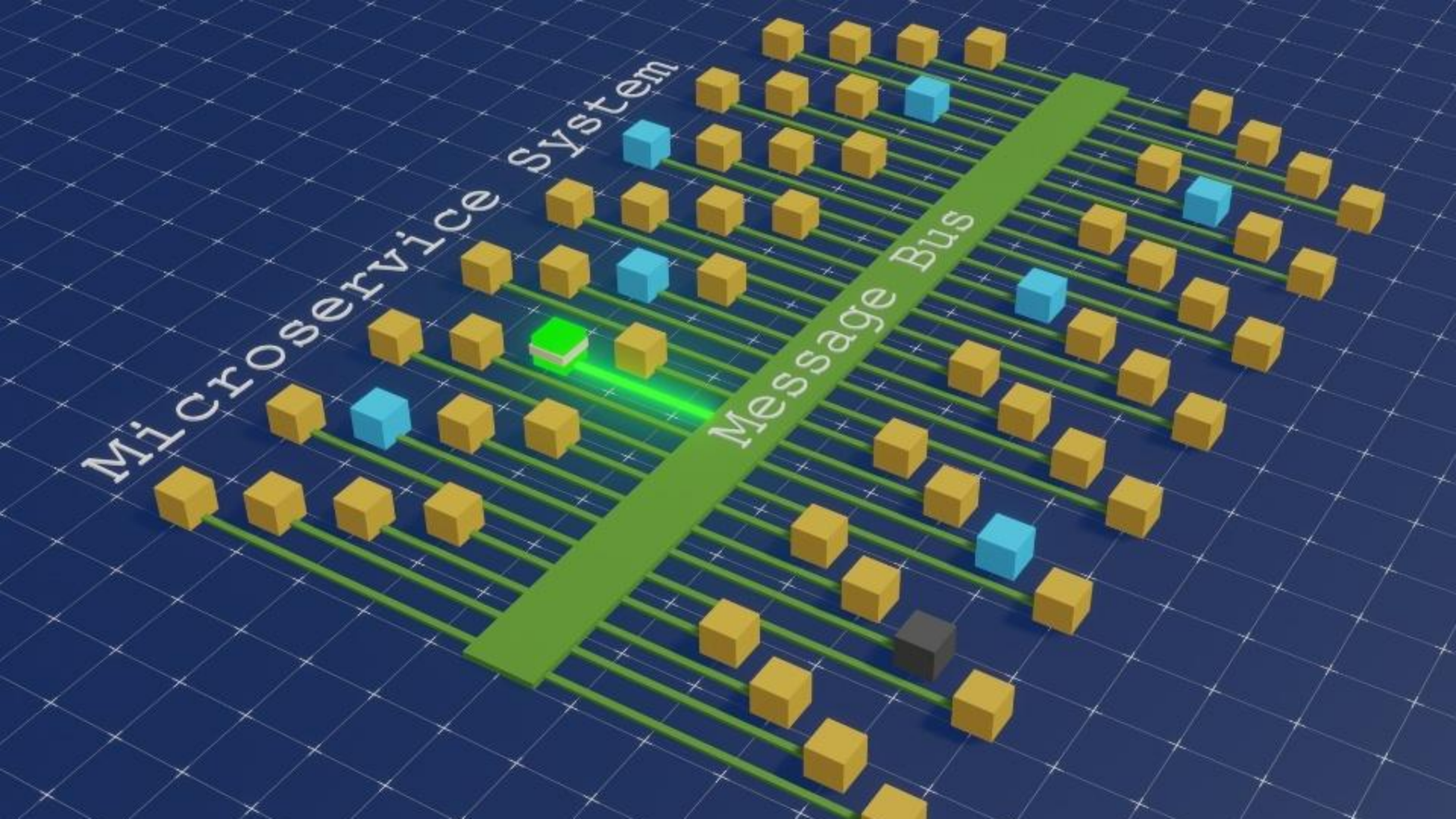

Microservice System
Message Bus

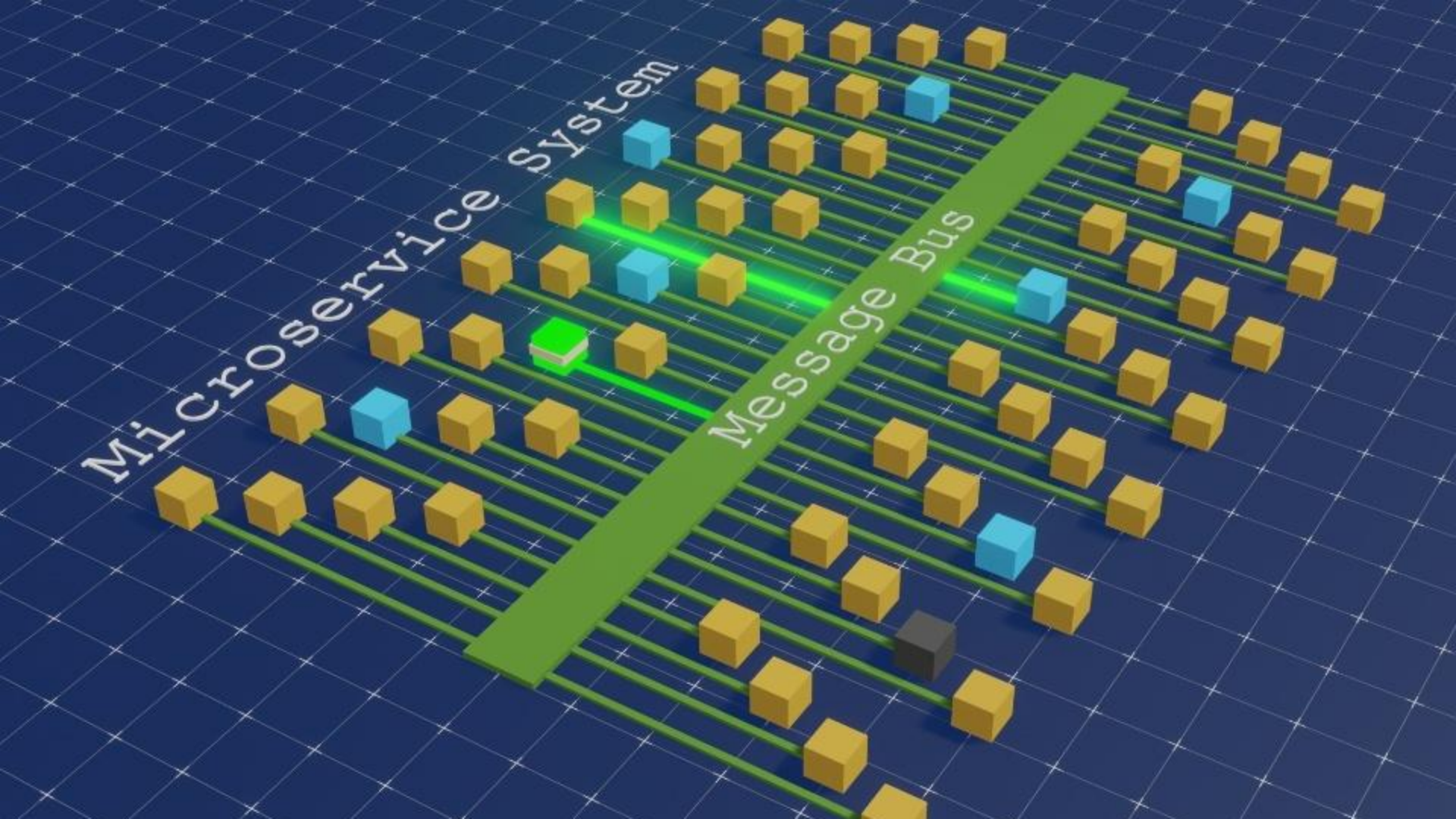

Microservice System
Message Bus

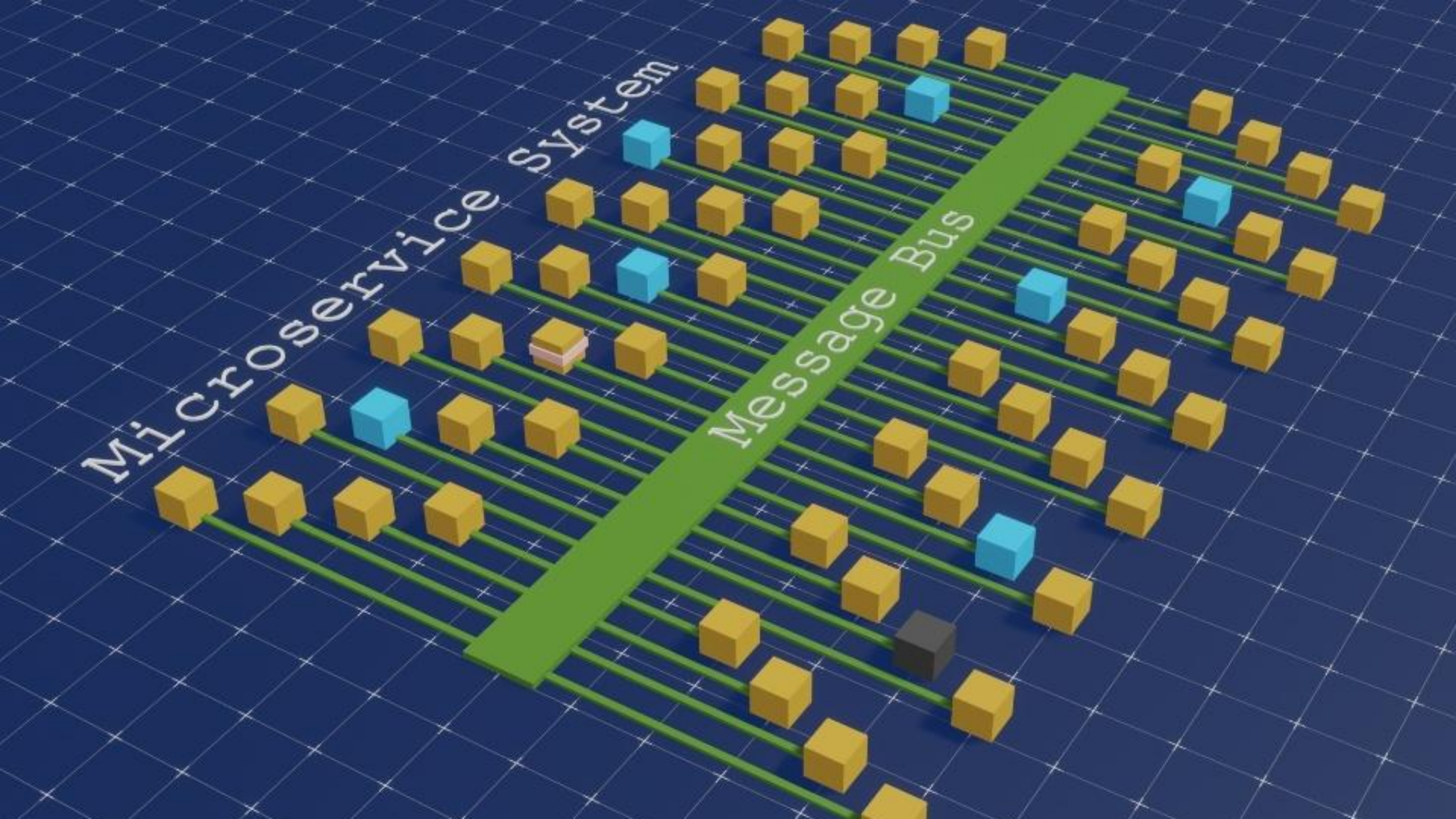

Microservice System
Message Bus

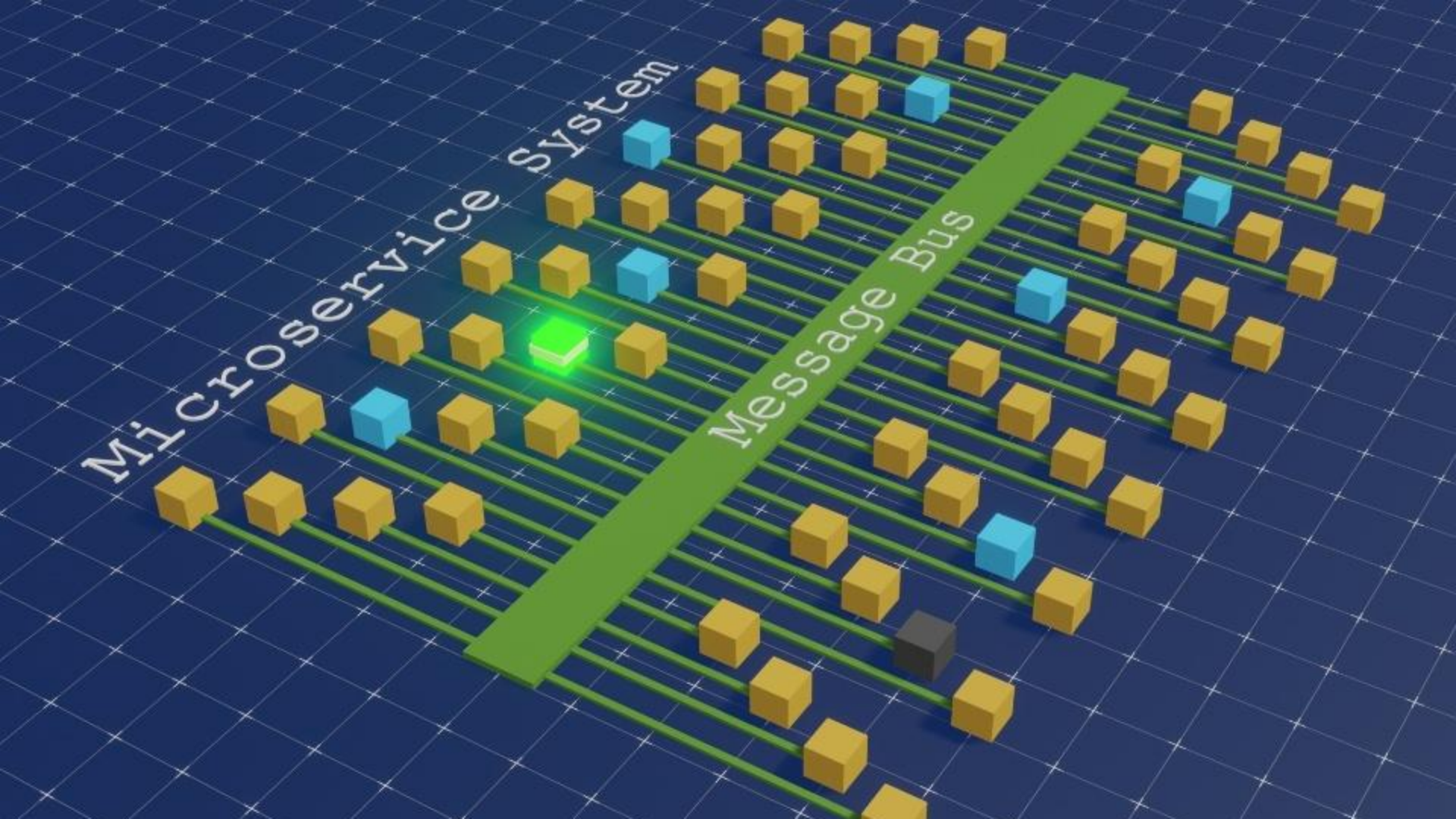

Microservice System
Message Bus

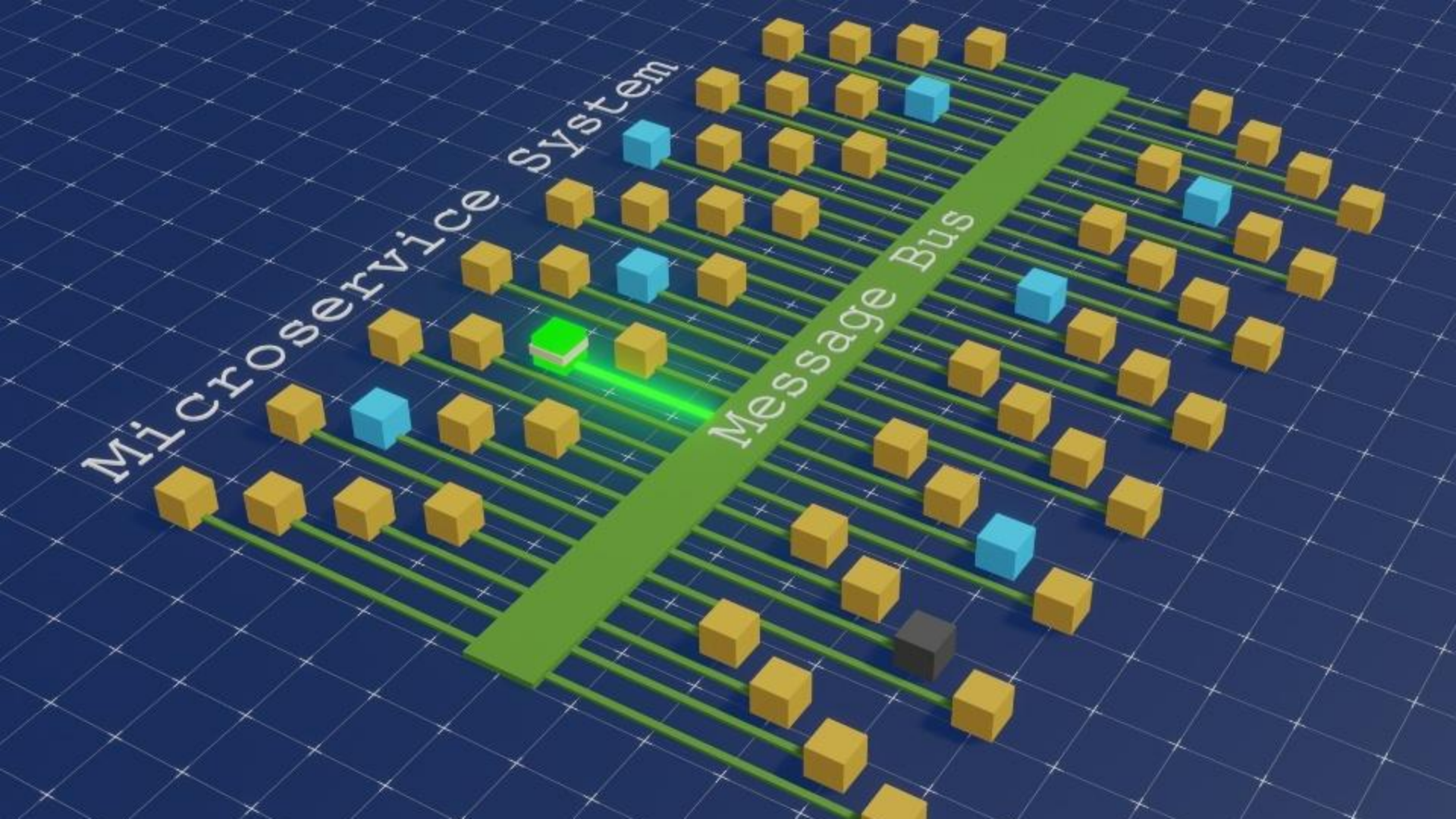

Microservice System
Message Bus

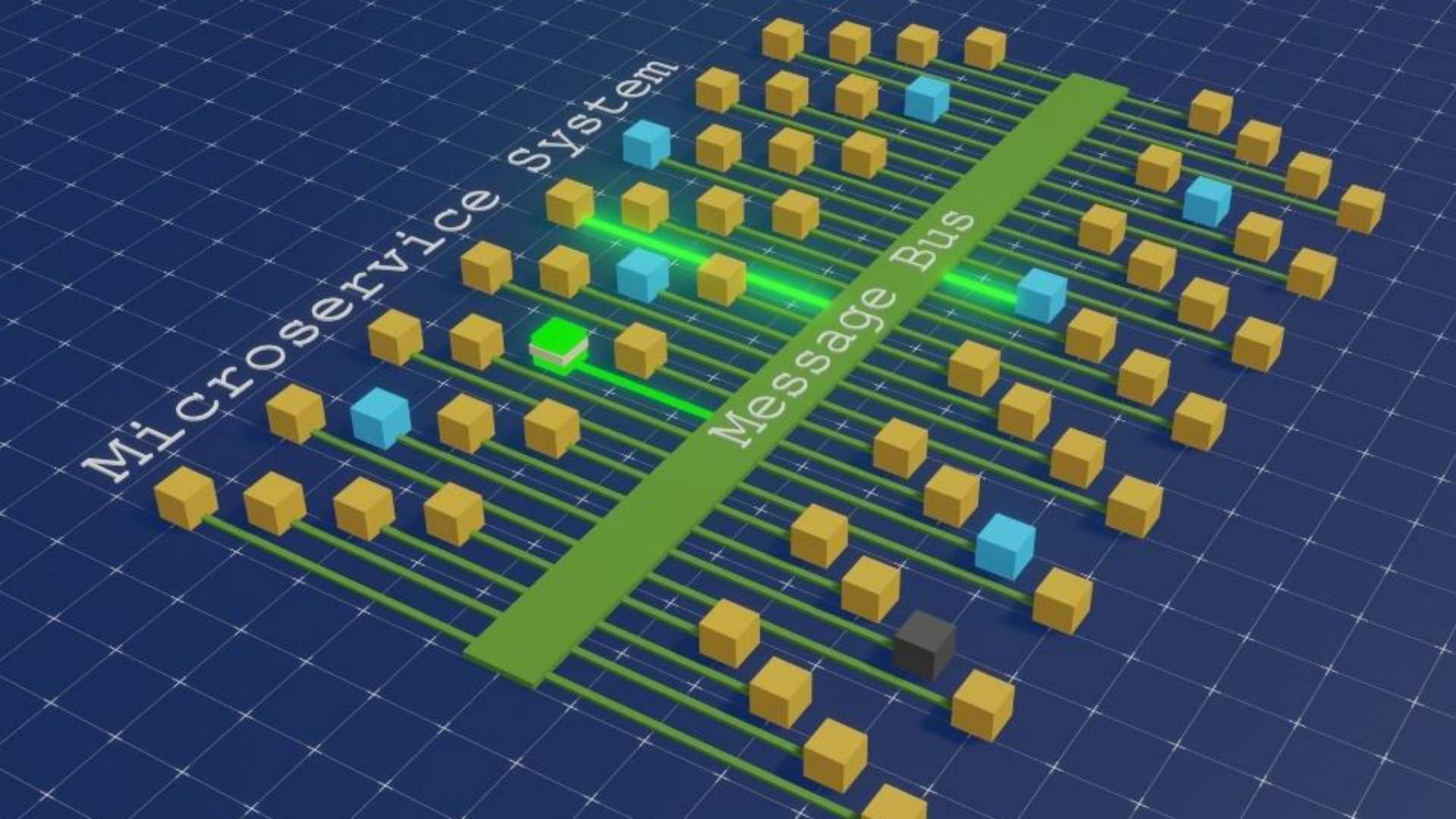

Microservice System
Message Bus

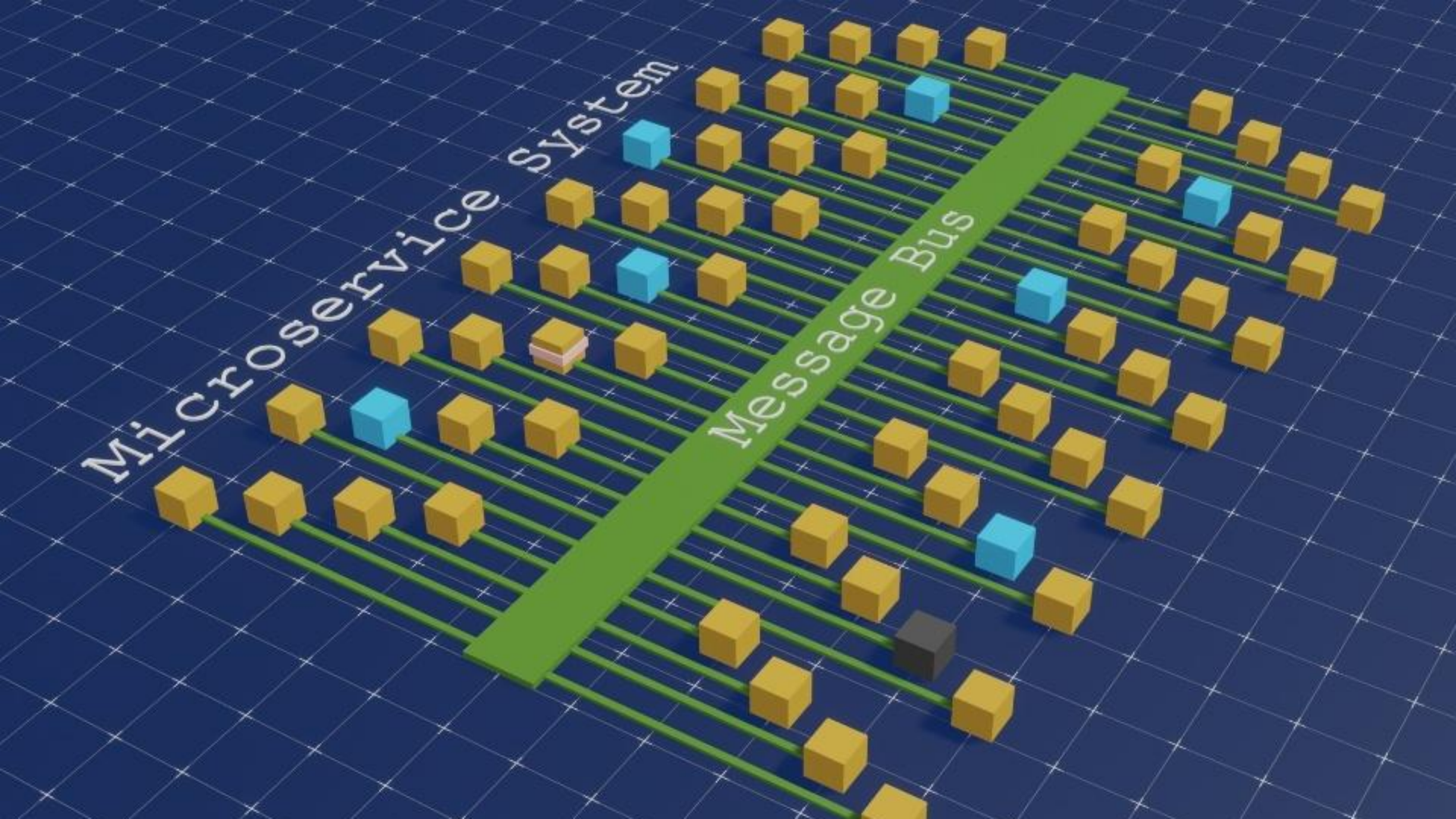

Microservice System
Message Bus

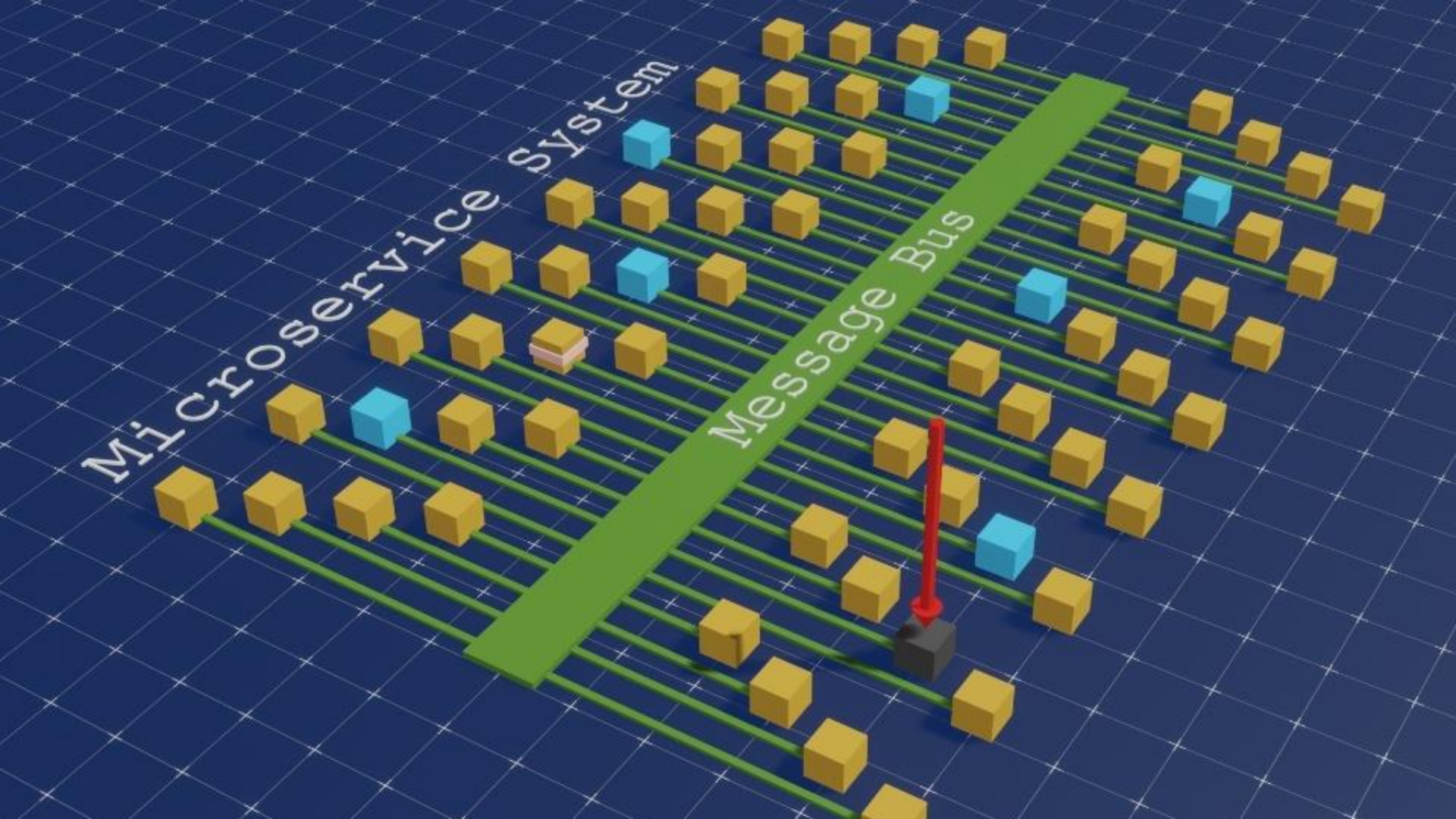

Microservice System
Message Bus

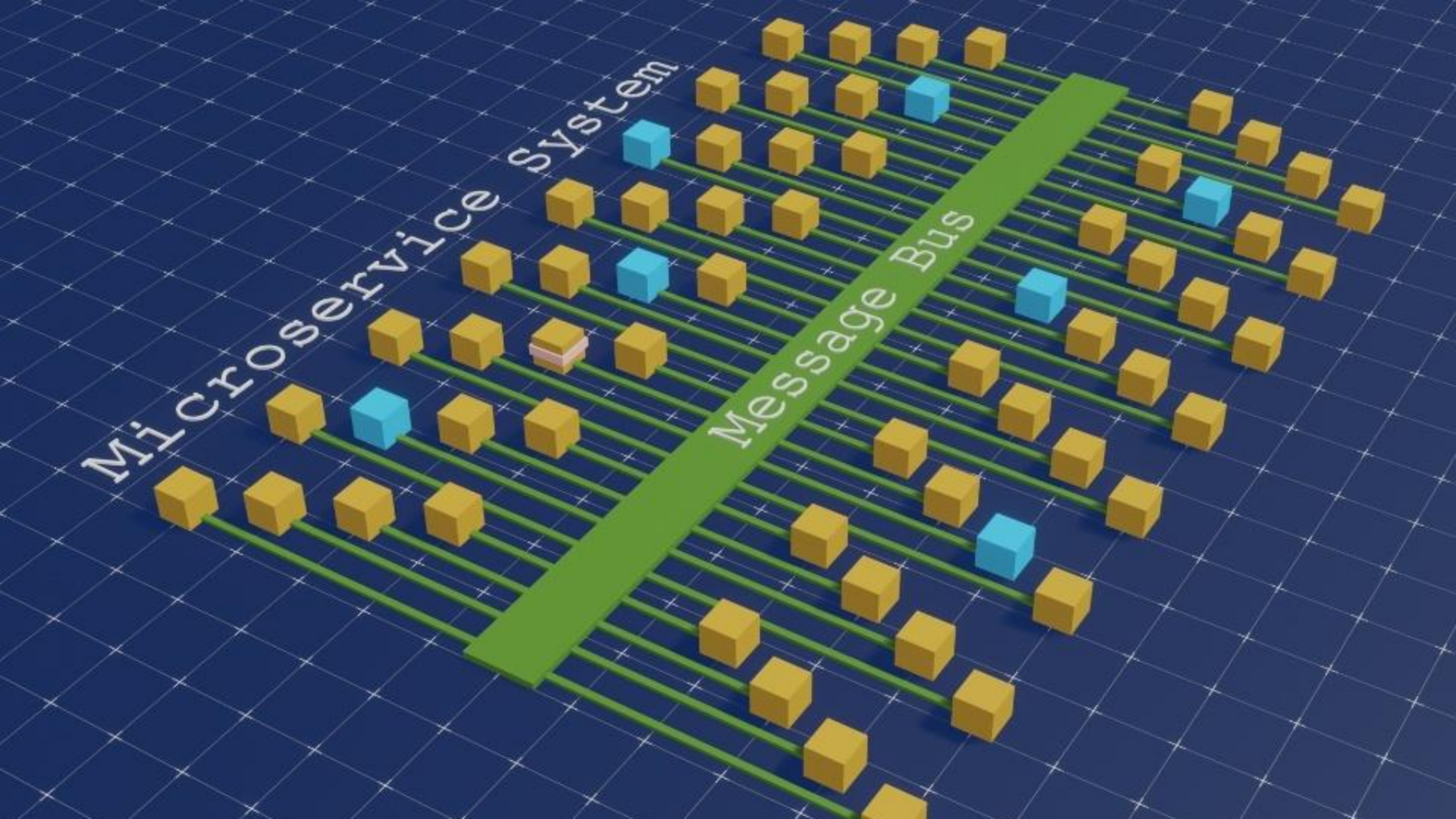

Microservice System
Message Bus

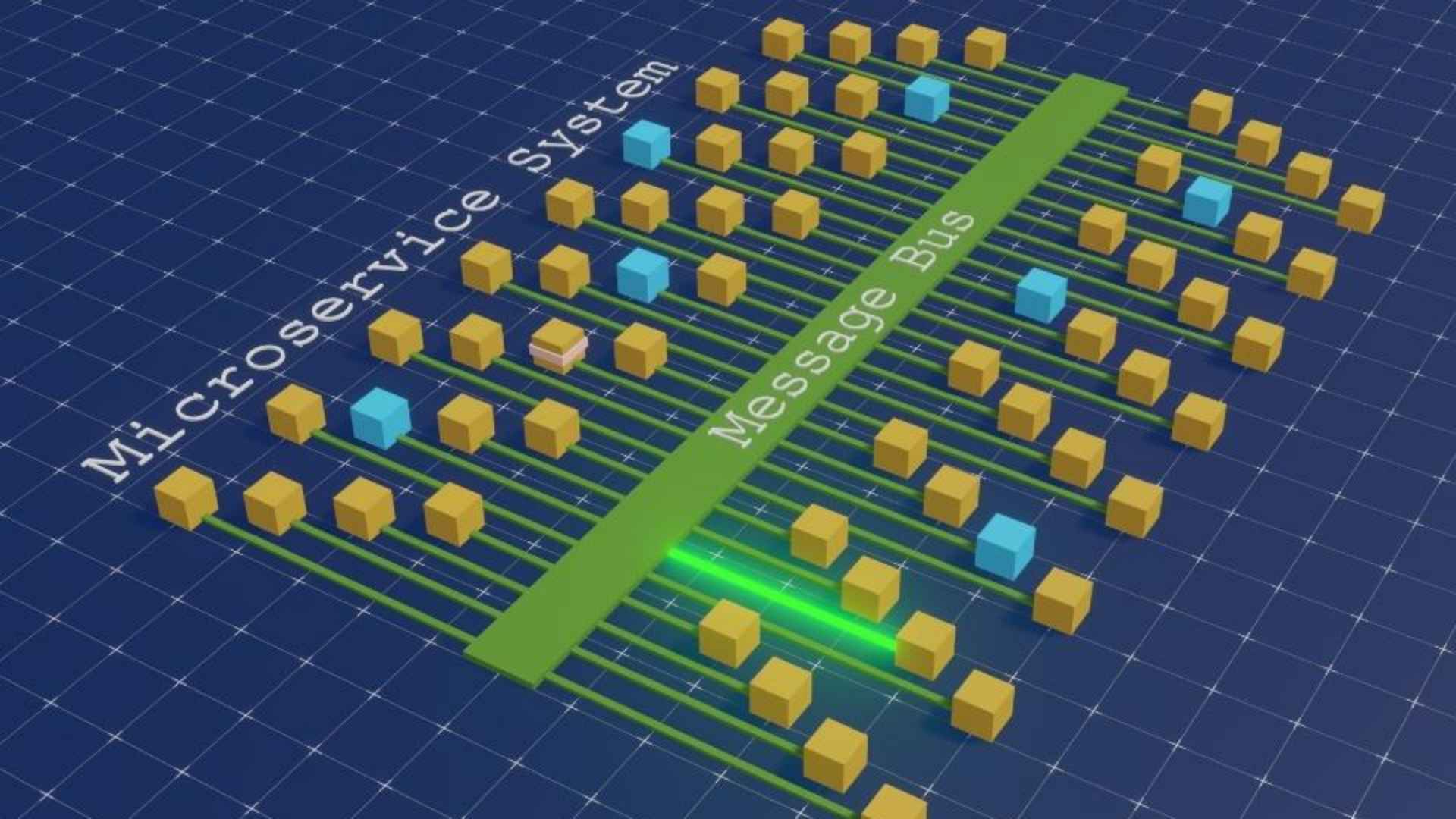

Microservice System
Message Bus

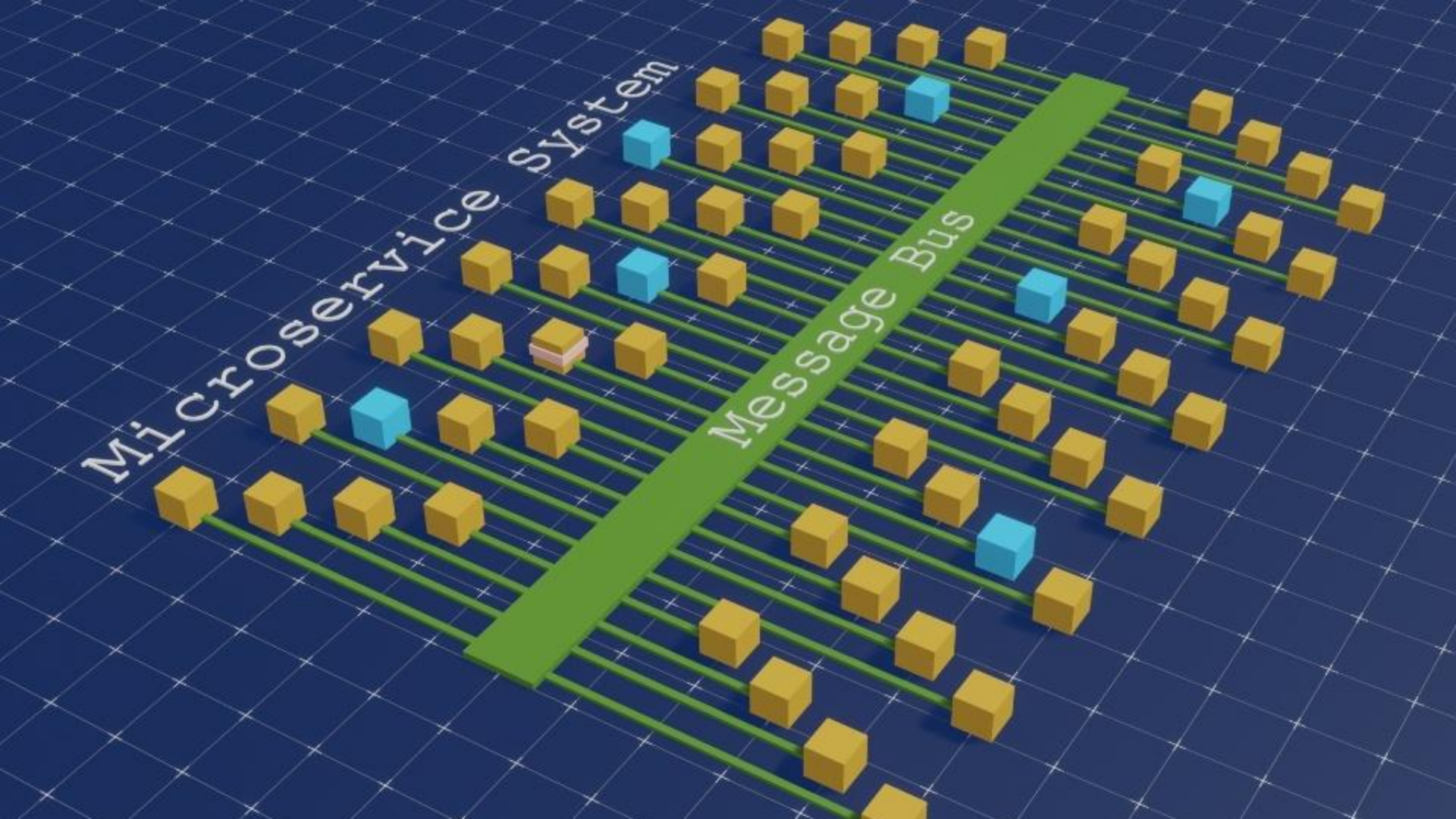

Microservice System
Message Bus

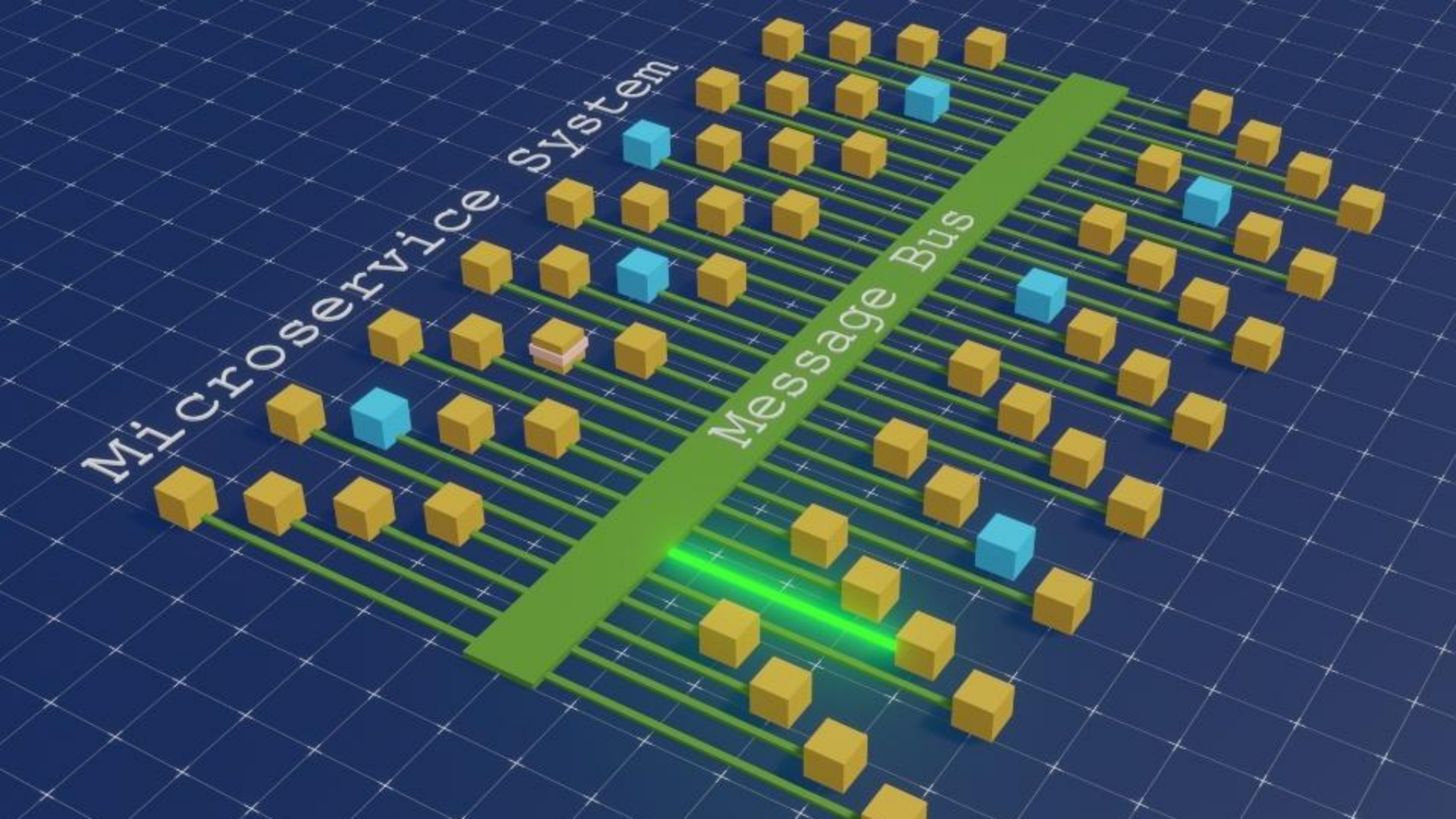

Microservice System
Message Bus

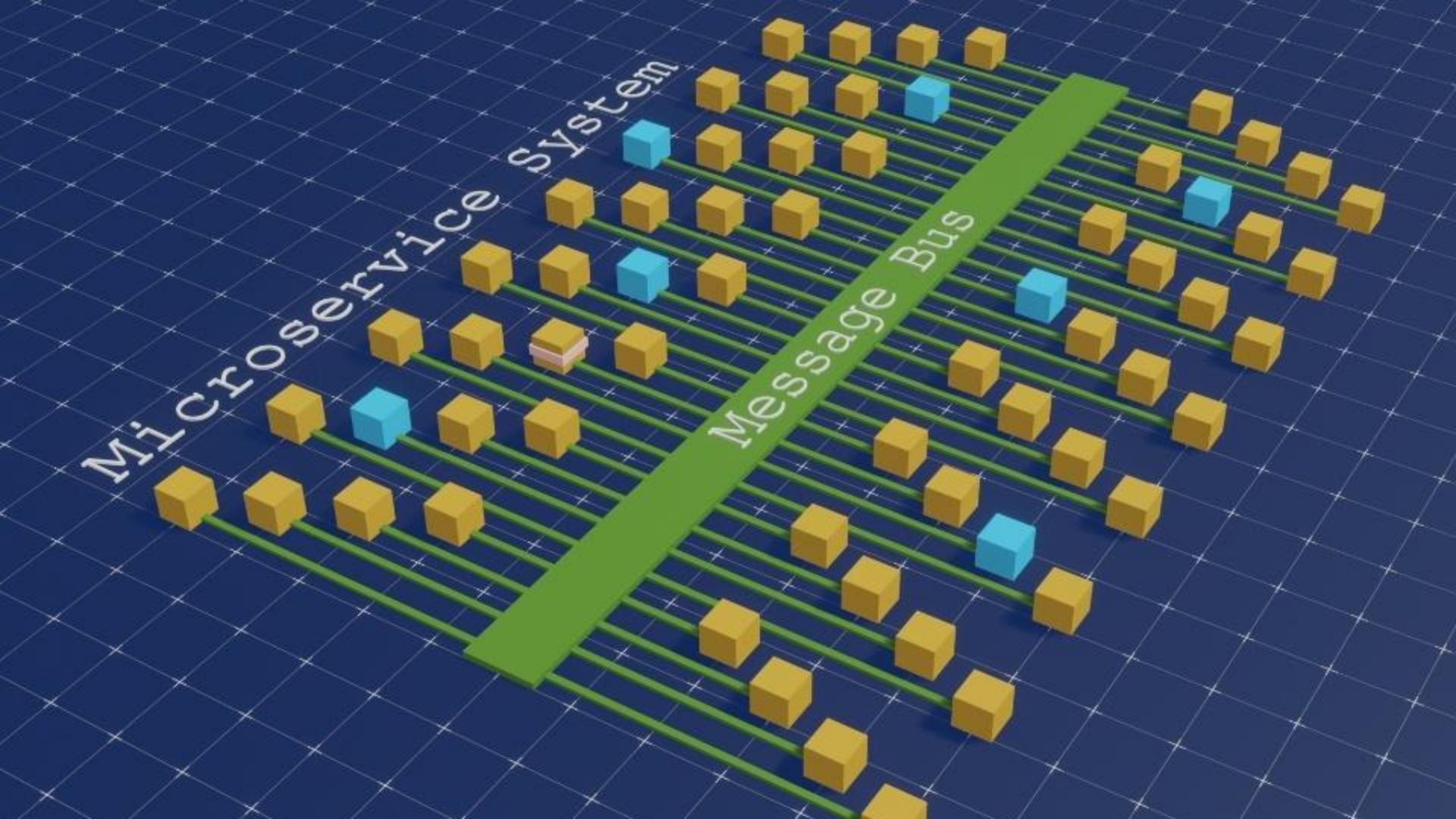

Microservice System
Message Bus

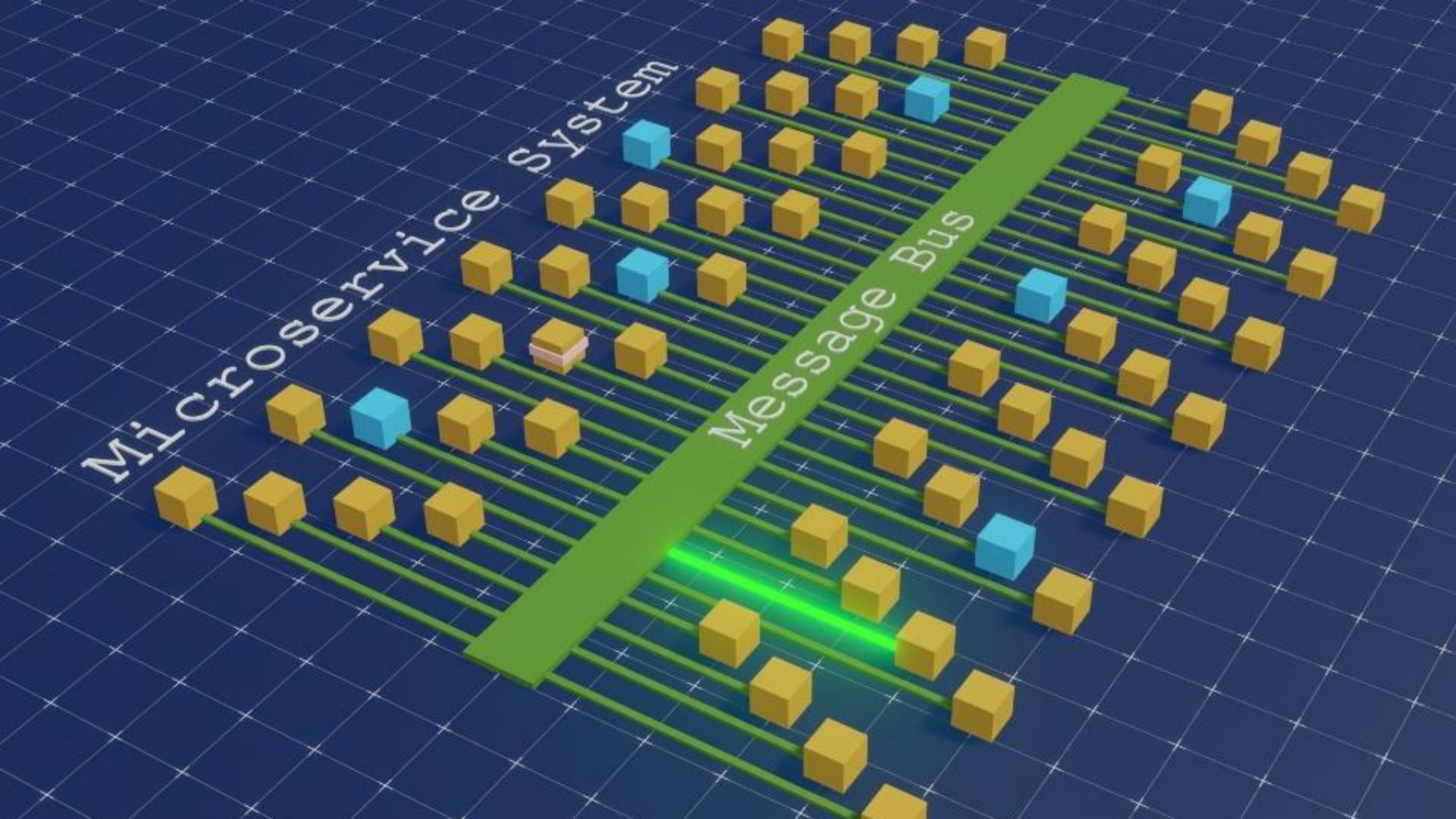

Microservice System
Message Bus

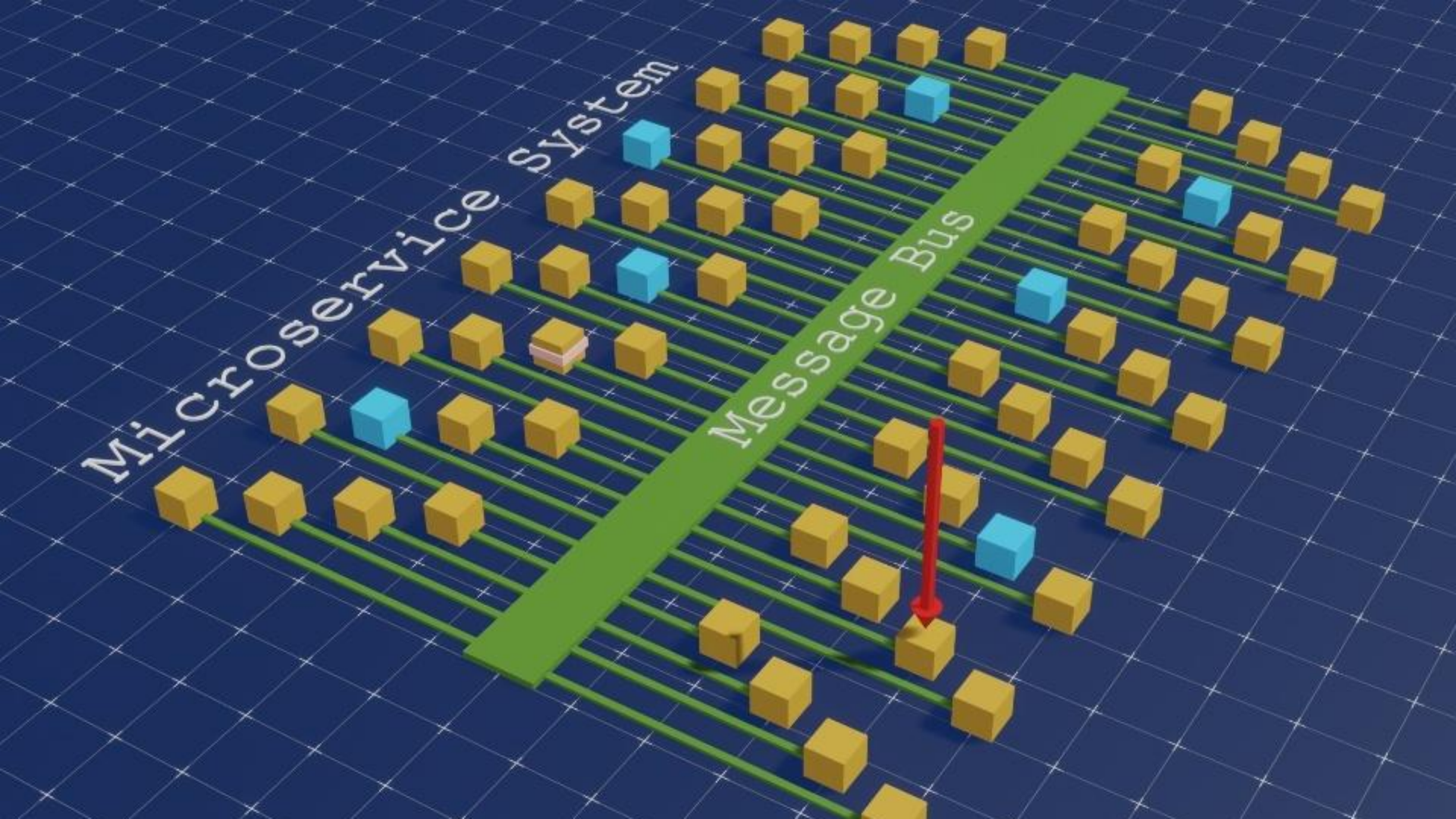

Microservice System
Message Bus

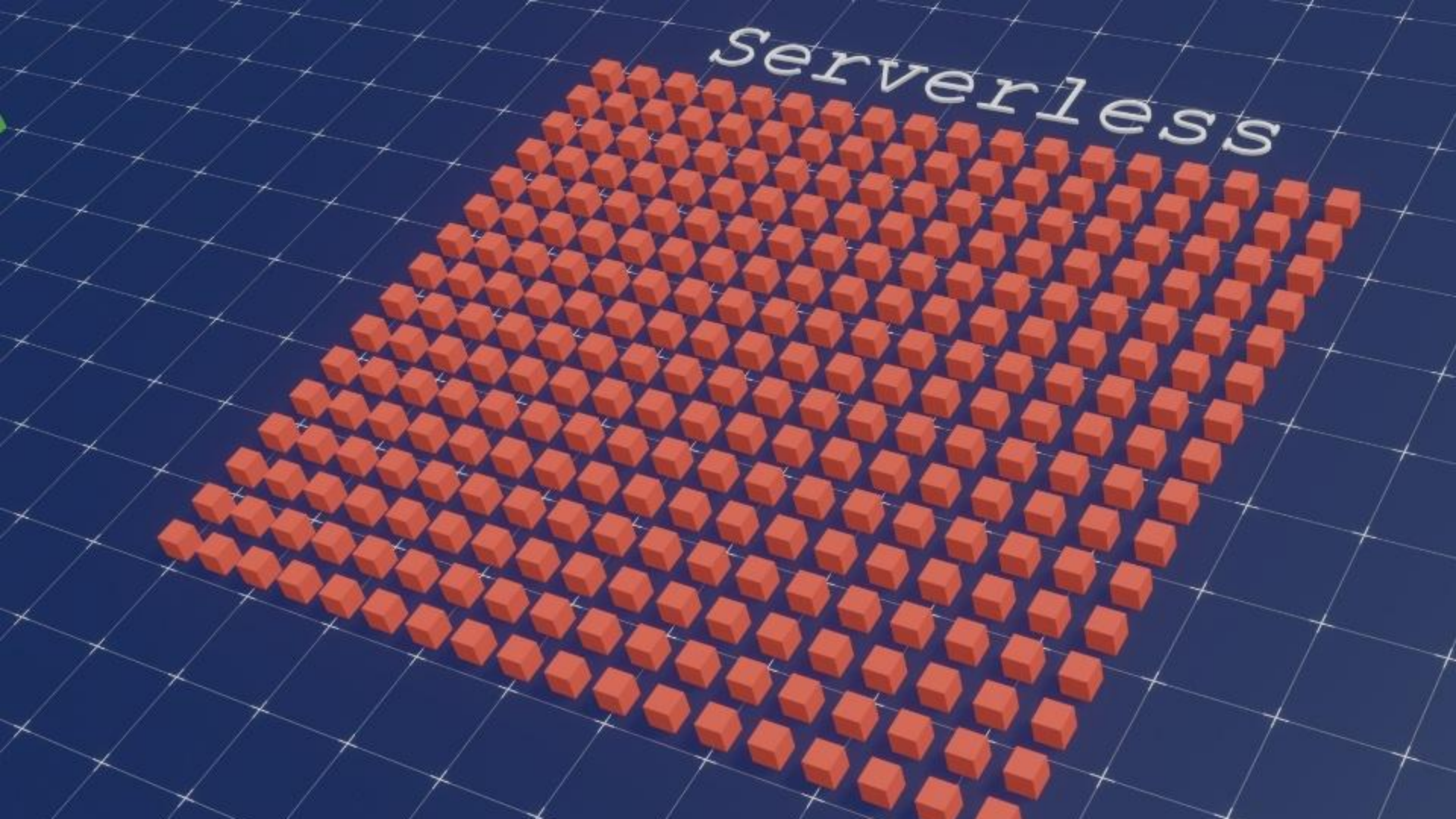

Microservice System
Message Bus

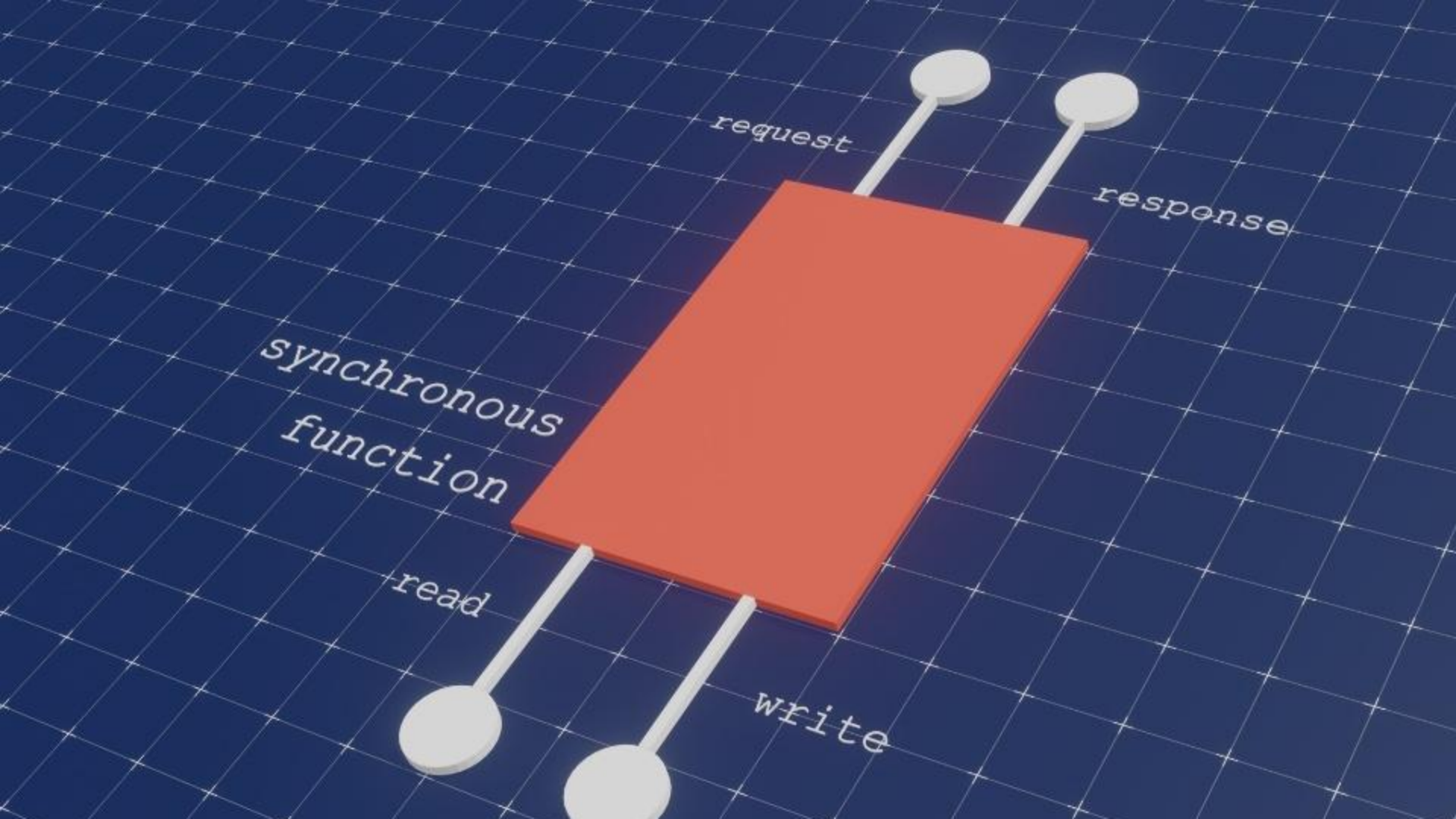

Microservice System
Message Bus

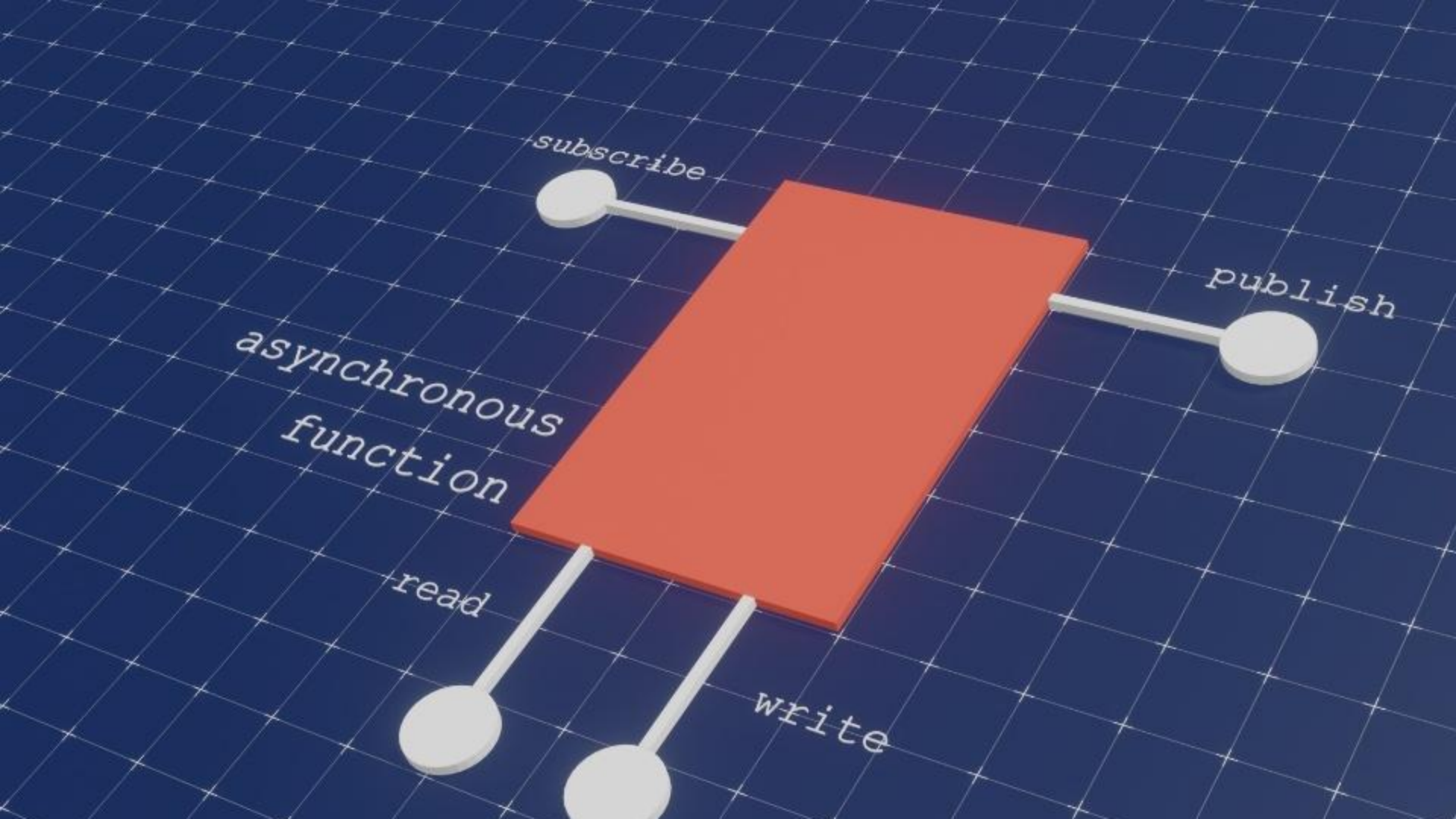

Microservice System
Message Bus

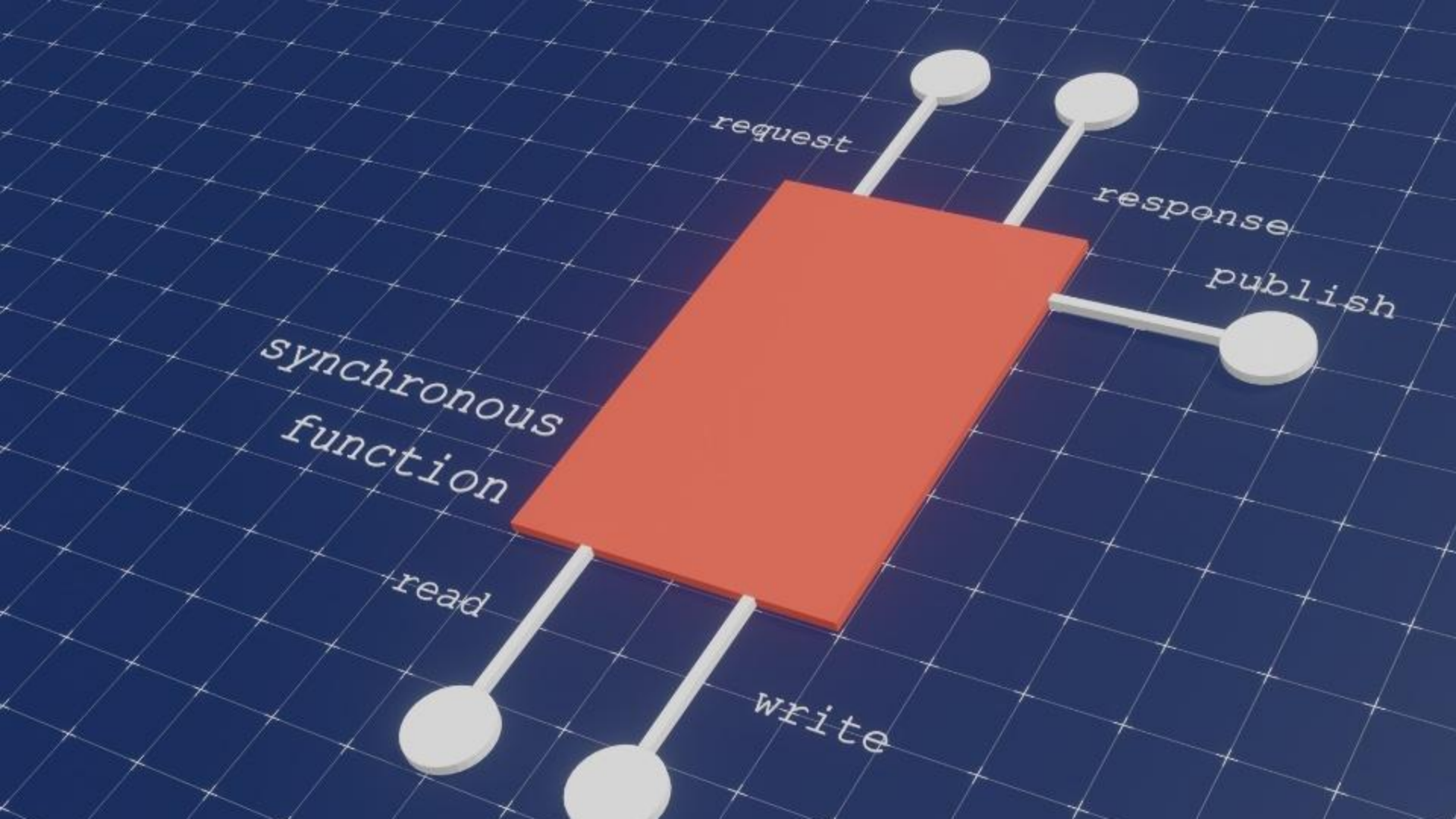

Microservice System
Message Bus

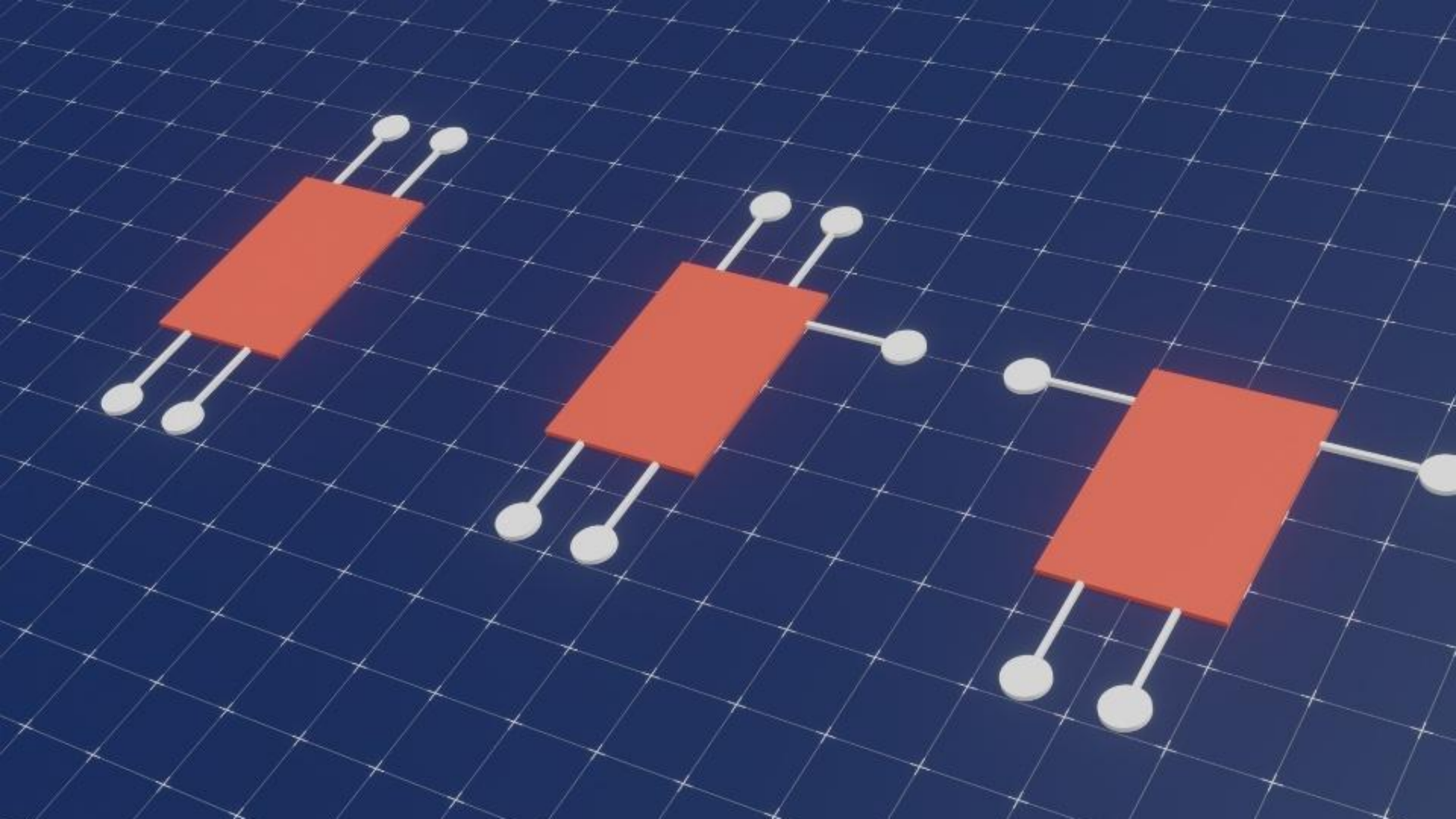

Microservice System
Message Bus

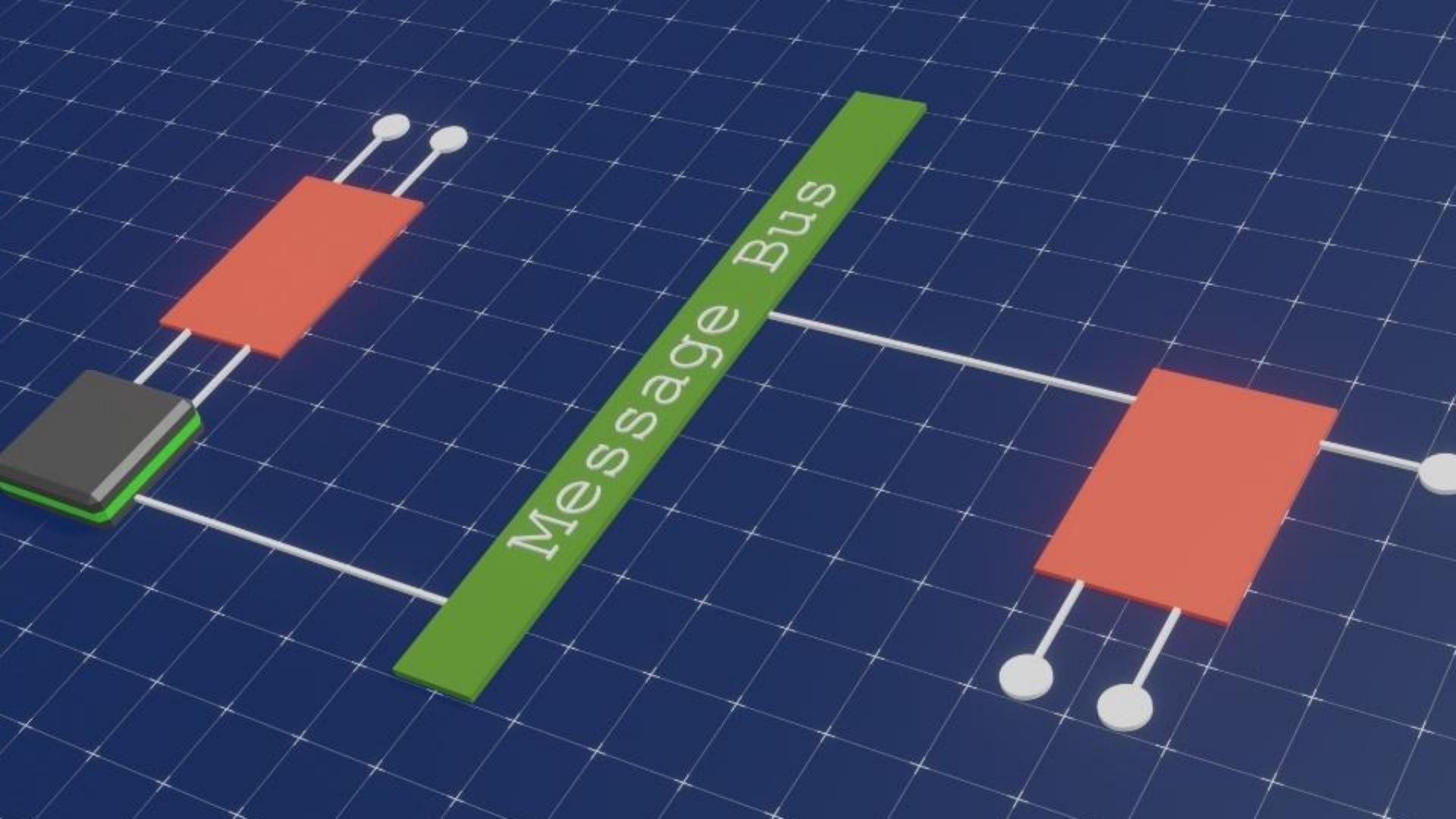

Serverless

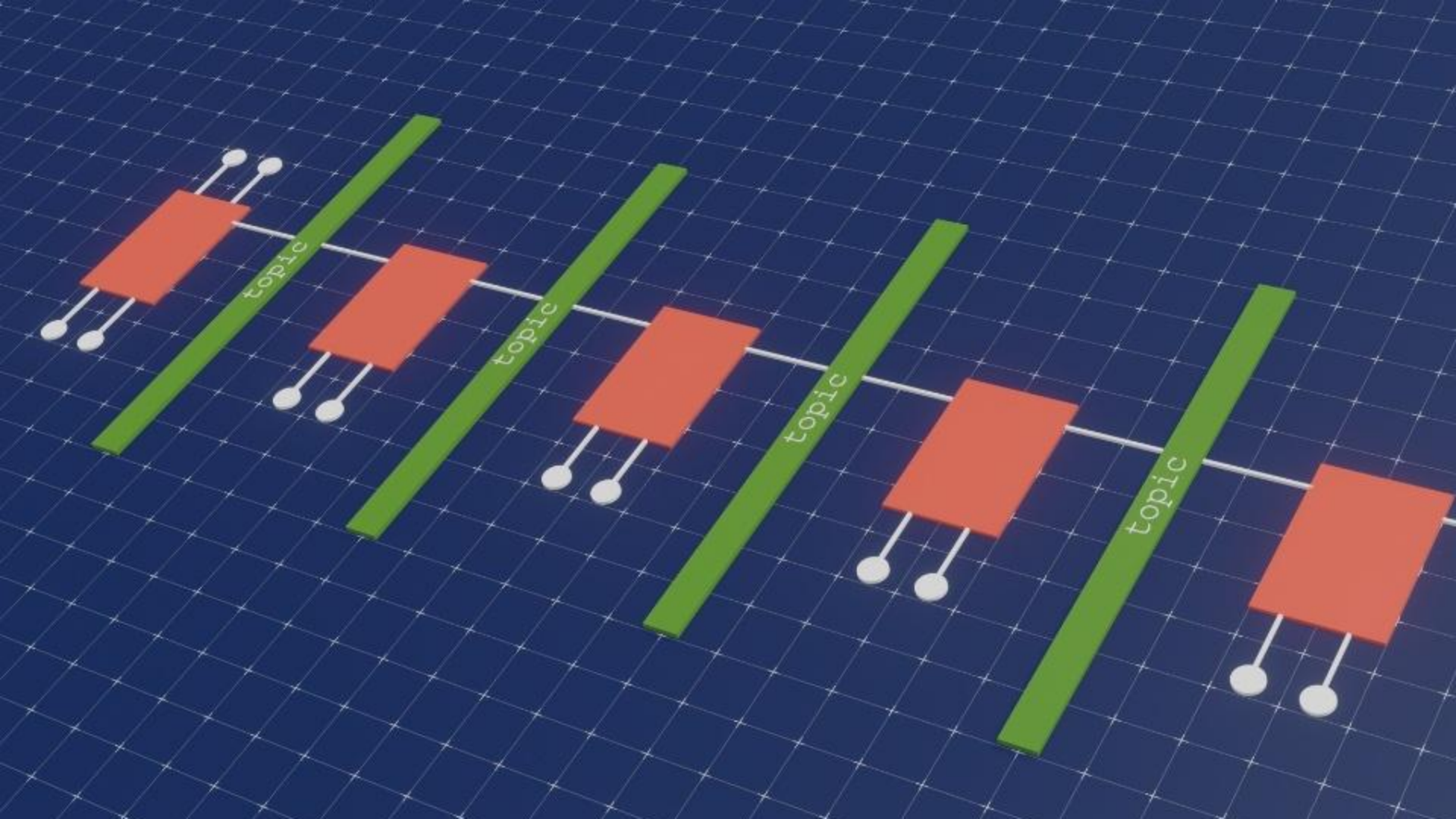

request
response
synchronous
function
read
write

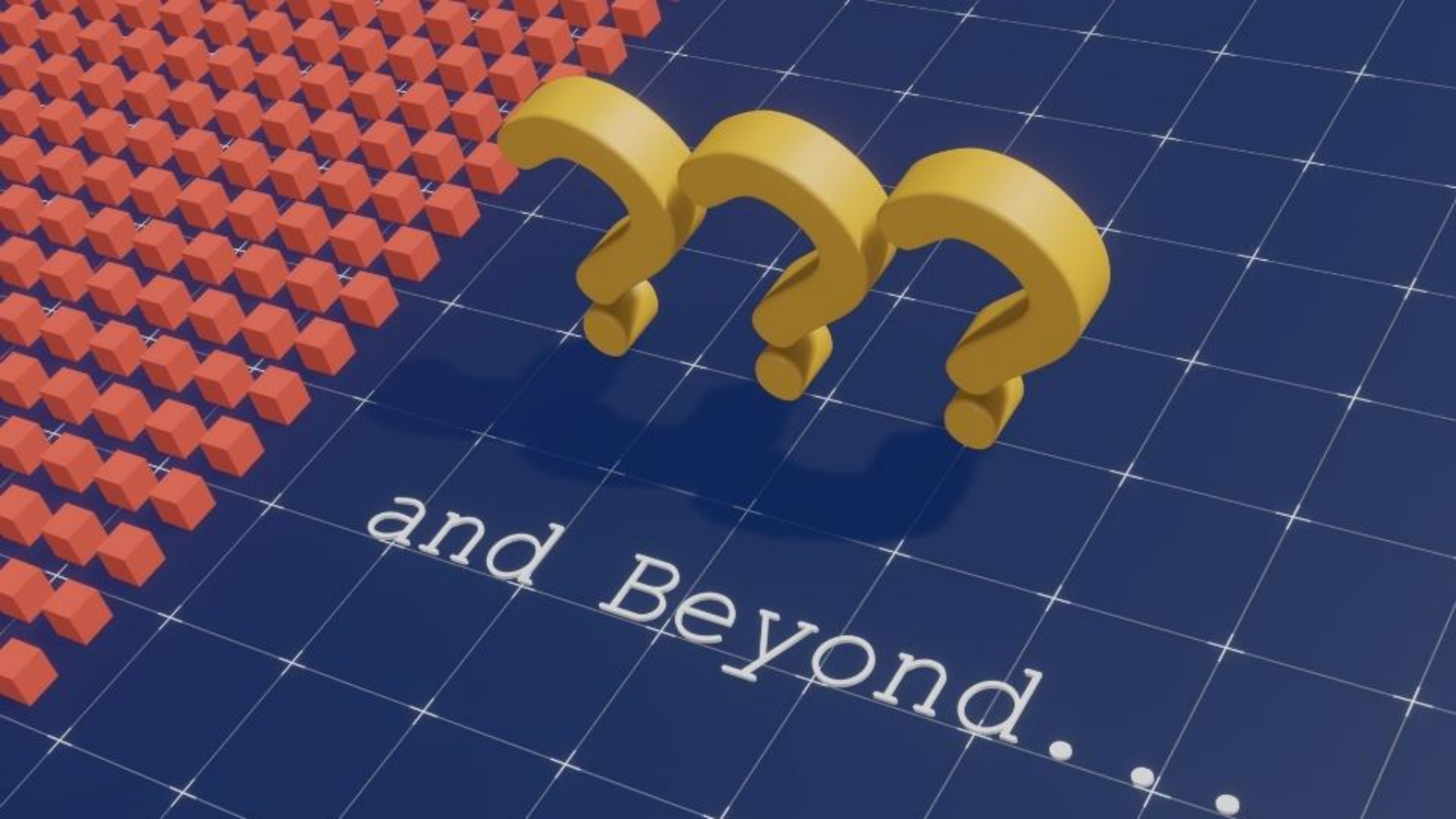

subscribe
publish
asynchronous
function
read
write

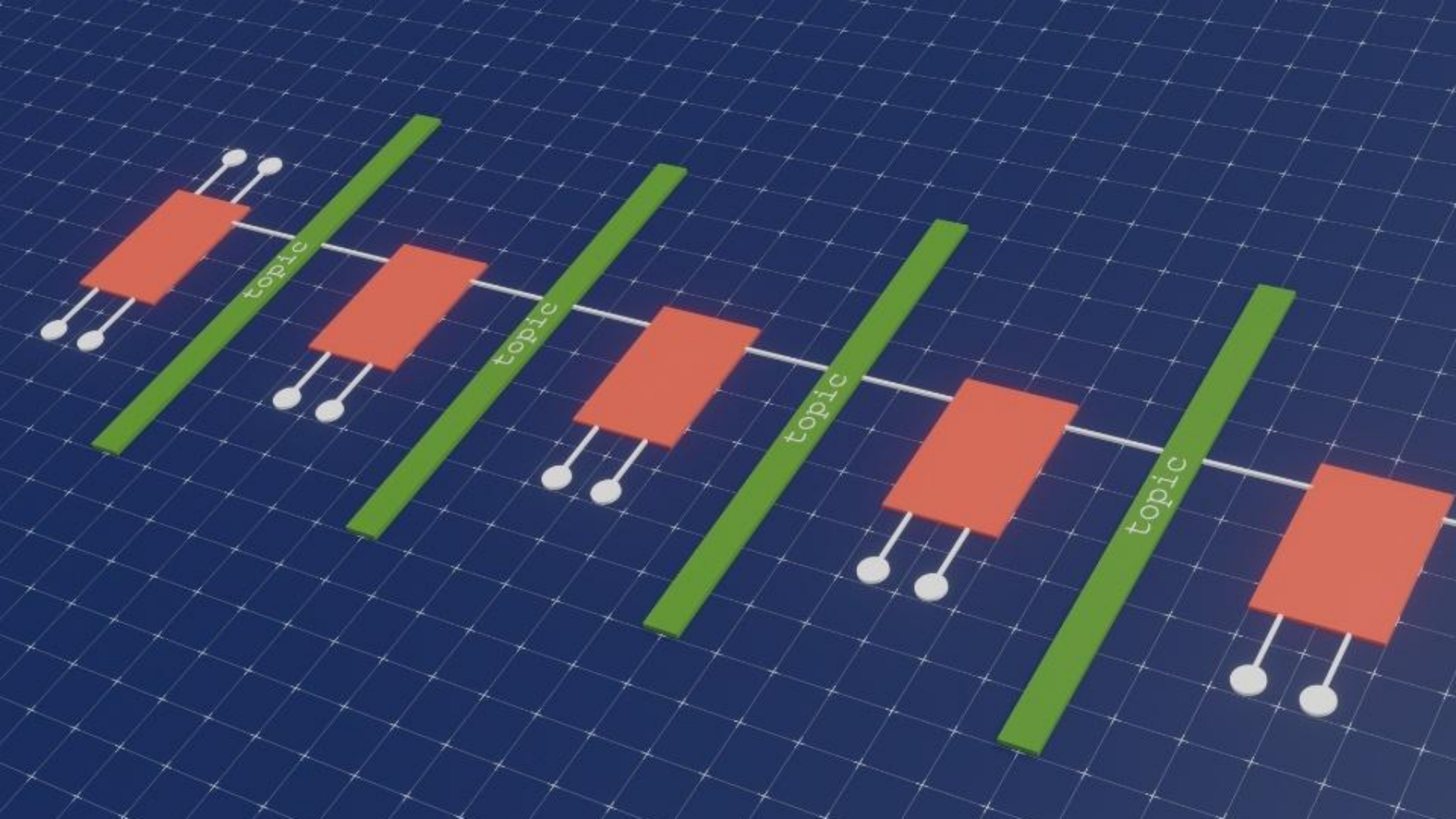

request
response
publish
synchronous
function
read
write

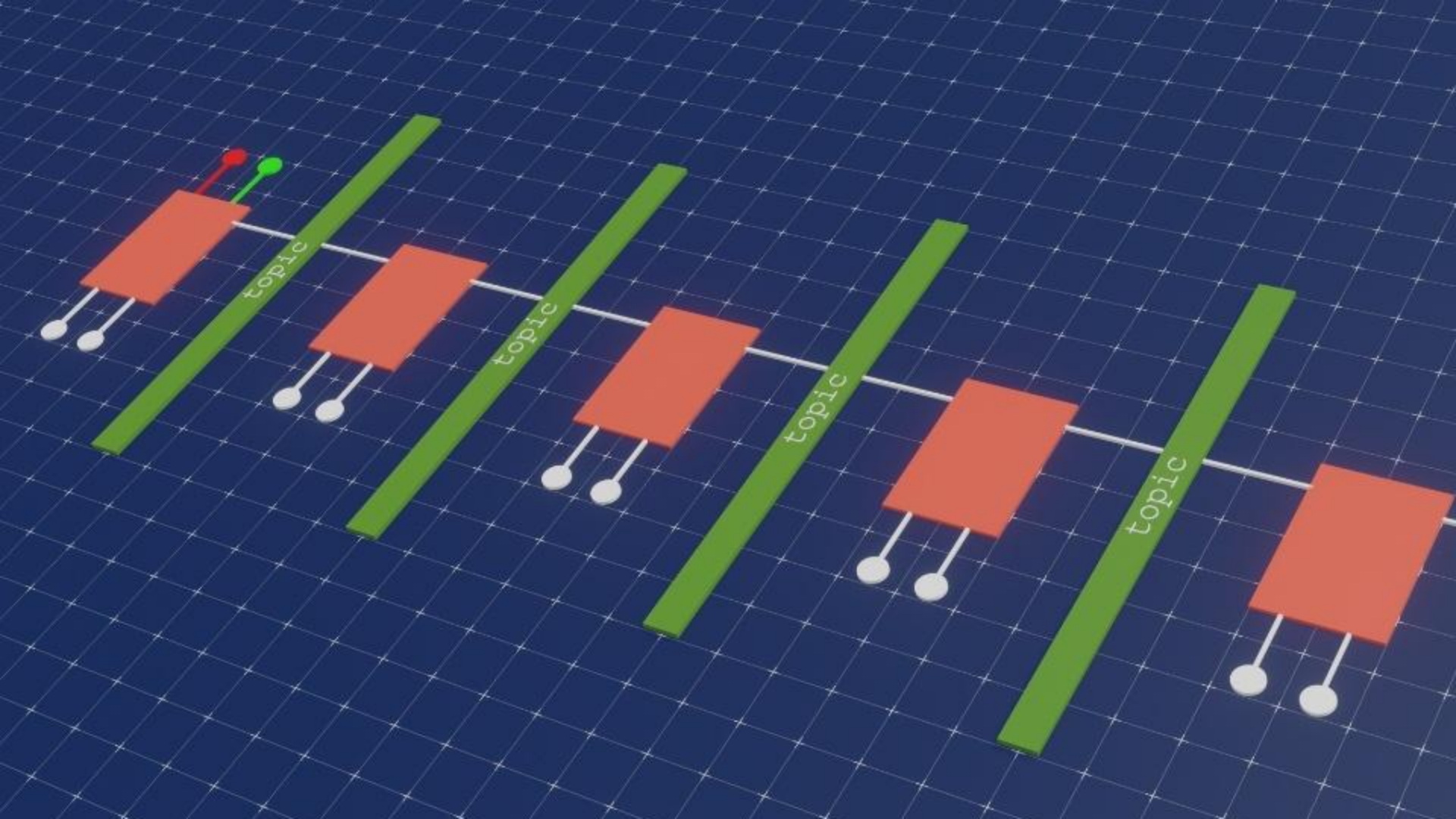


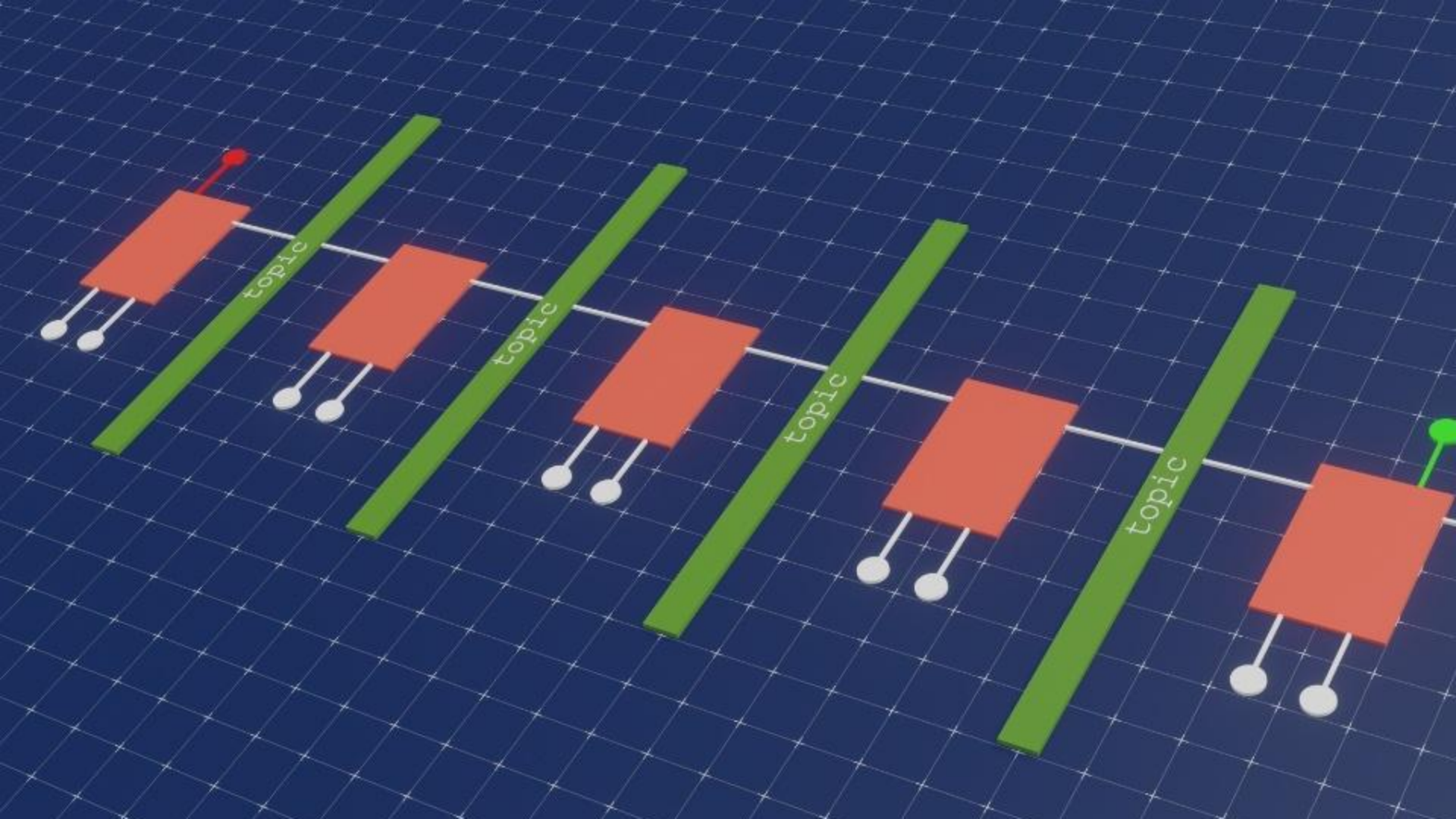

Message Bus

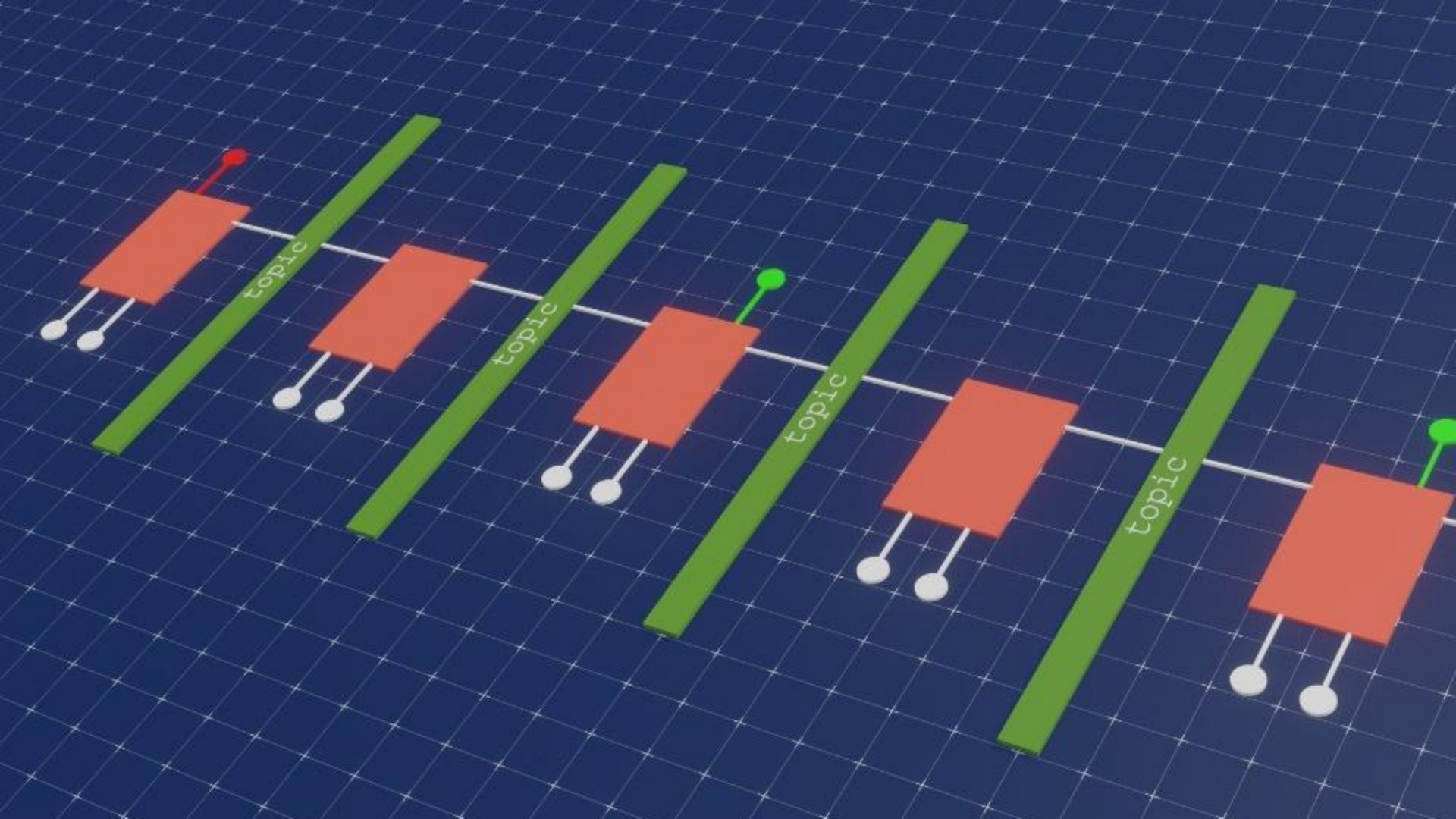

topic
topic
topic
topic

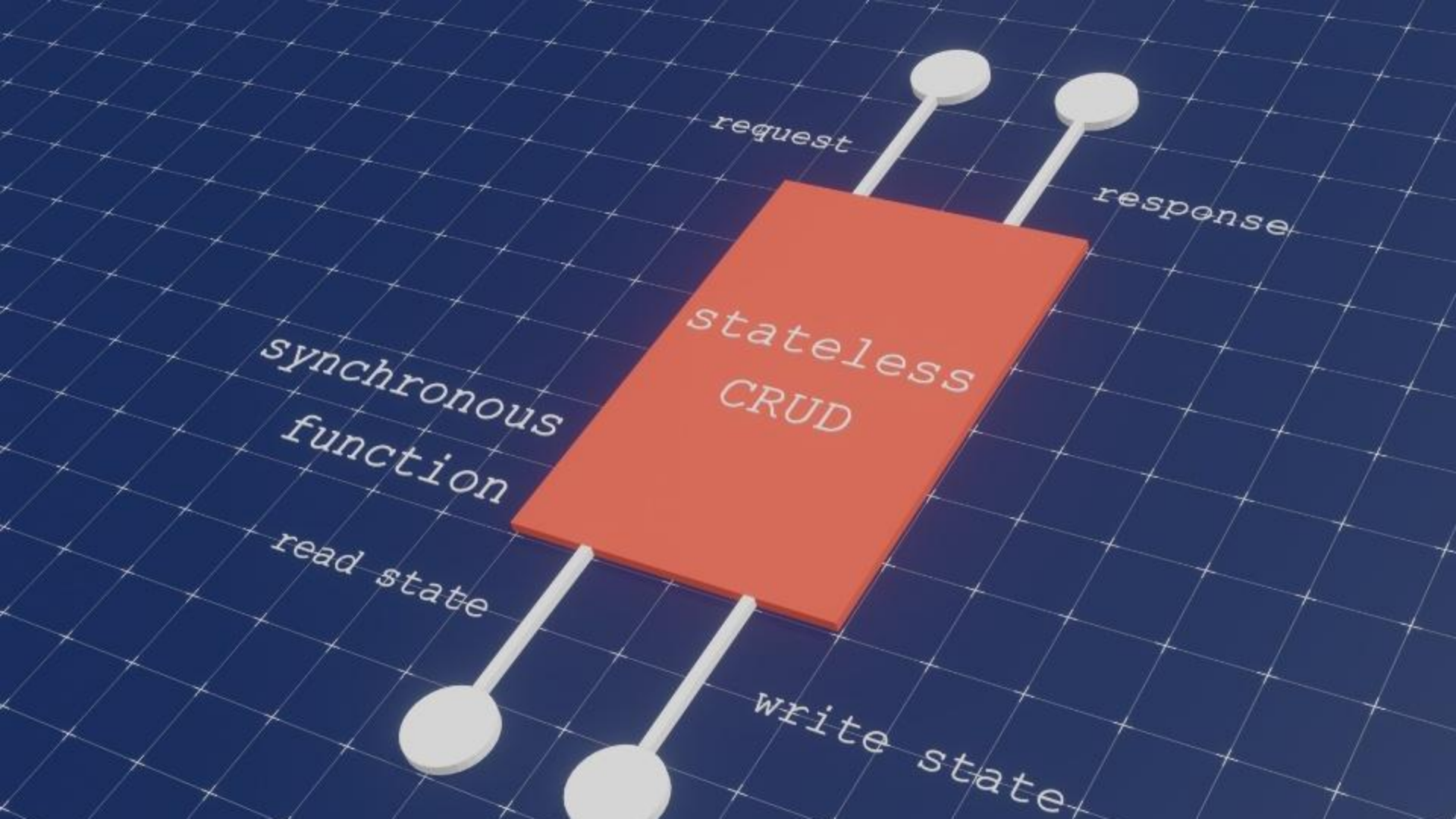

and Beyond...

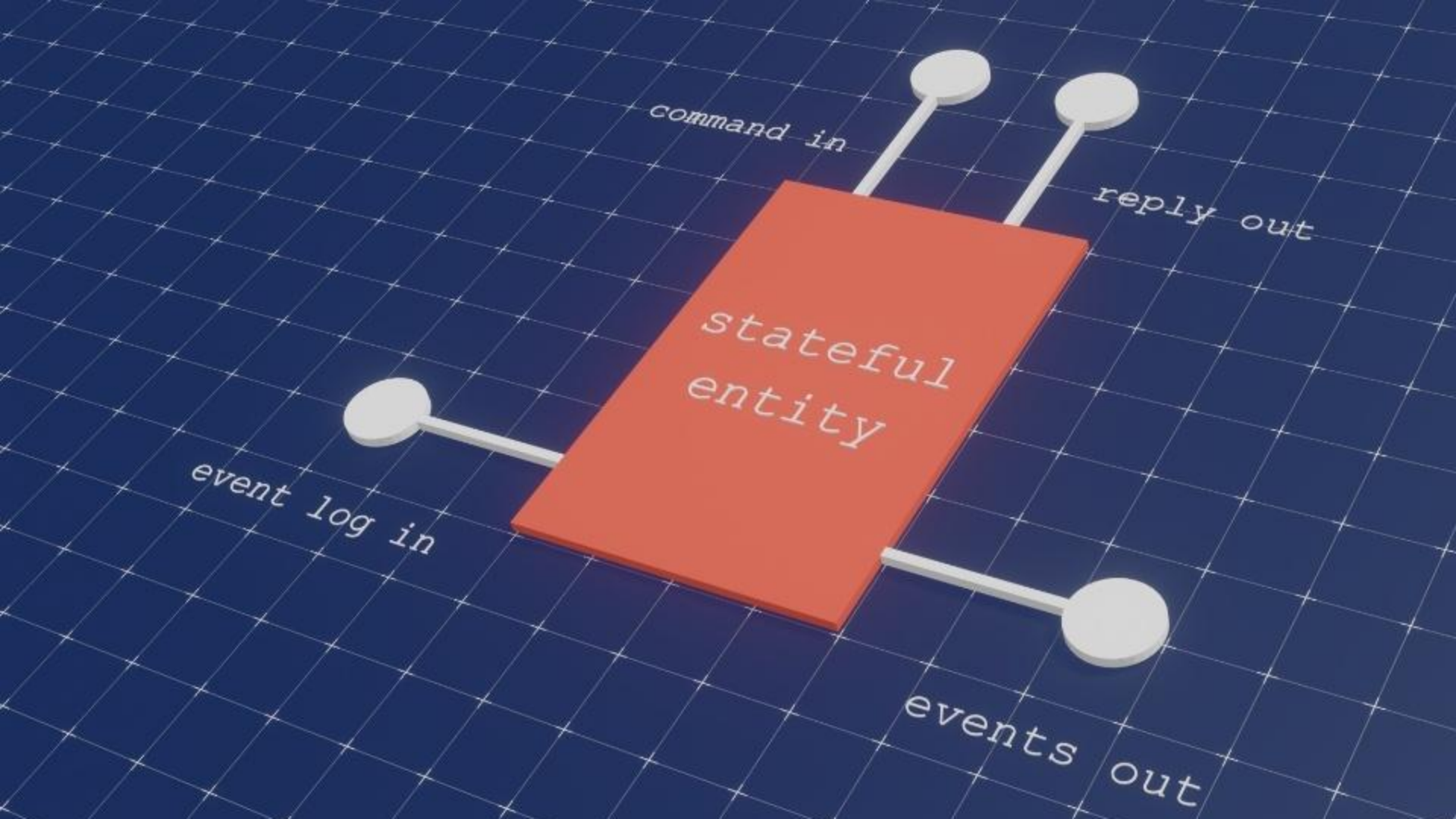

topic
topic
topic
topic

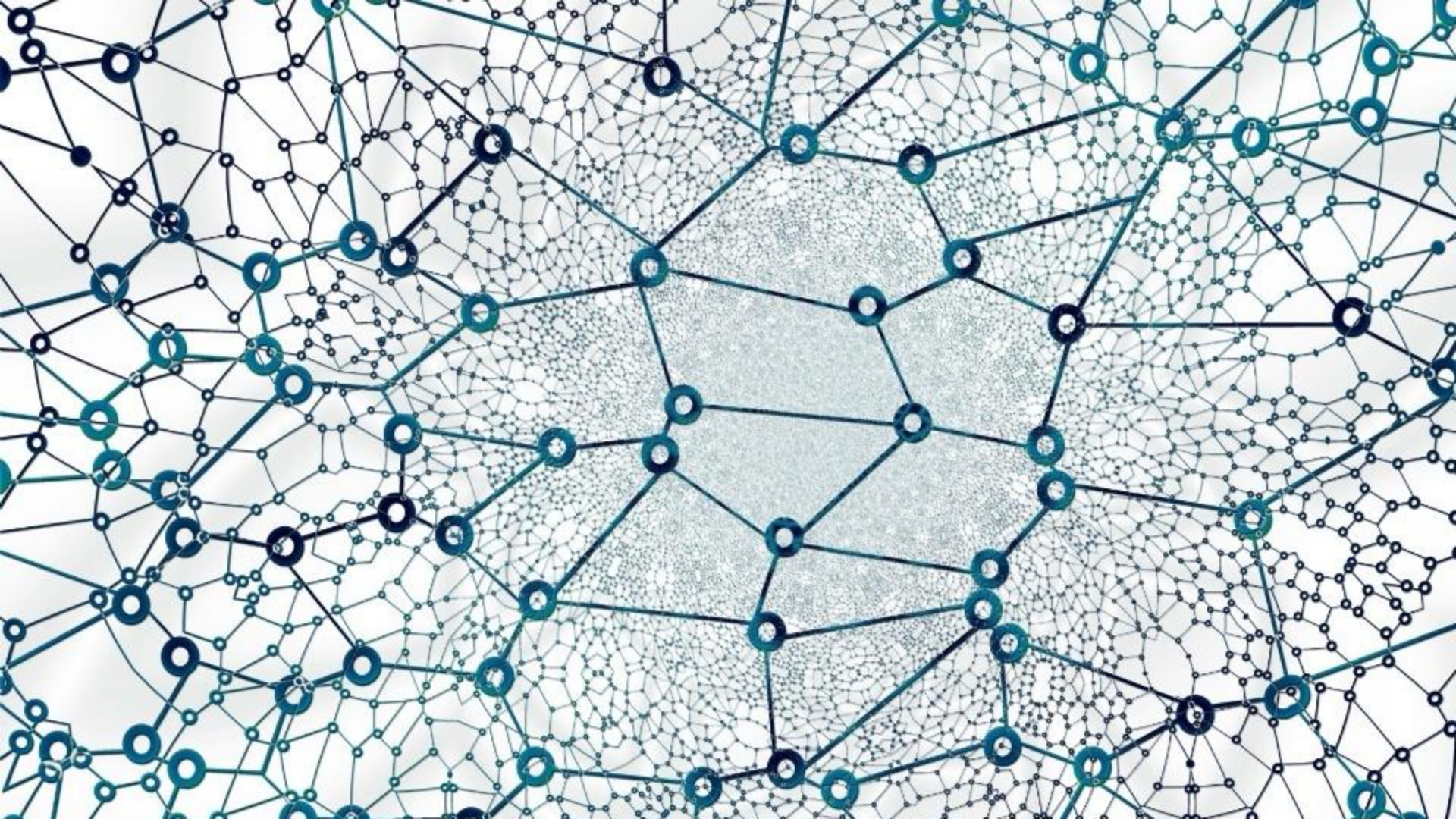

topic
topic
topic
topic

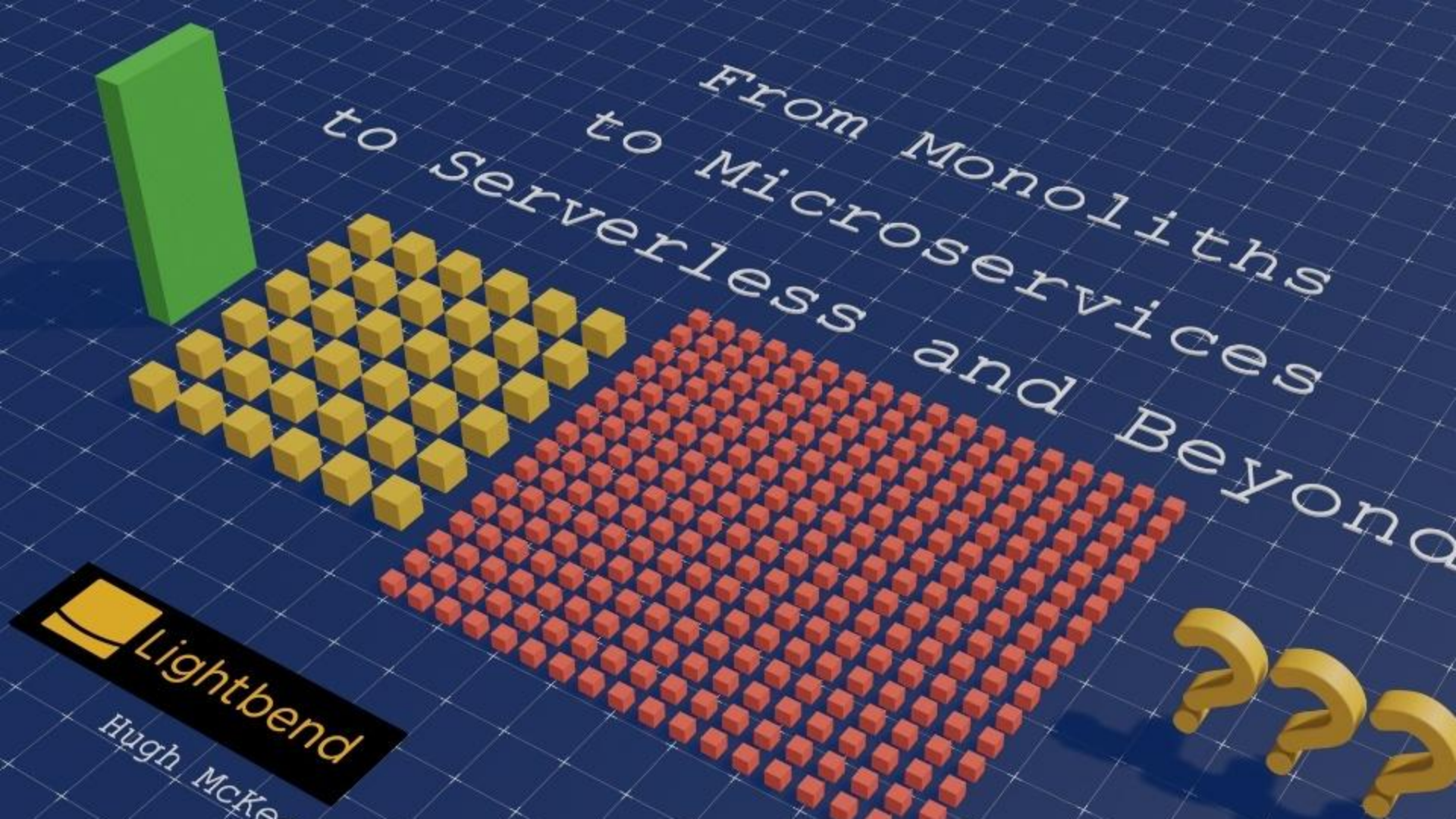

topic
topic
topic
topic


topic
topic
topic
topic


request
response
stateless
CRUD
synchronous
function
read state
write state


command in
reply out
stateful
entity
event log in
events out




From Monoliths
to Microservices
to Serverless and Beyond
Lightbend
Hugh McKe

cloudstate.io

statefun.io

dapr.io

https://bit.ly/2IwdTmw

thank you!

Hugh McKe

Challenges with microservices

- Synchronous communication
- Keeping the configuration up to date
- It's hard to track a request
- Analyzing the usage of hardware resources on a component level
- Management of many small components can become costly and error-prone

The 8 fallacies of distributed envi

1. The network is reliable
2. Latency is zero
3. Bandwidth is infinite
4. The network is secure
5. Topology doesn't change
6. There is one administrator
7. Transport cost is zero
8. The network is homogeneous

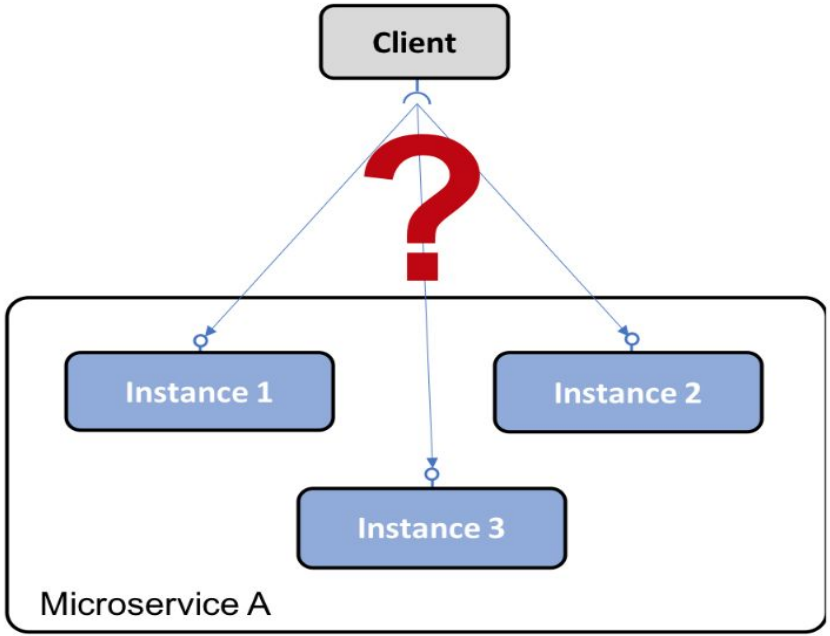
https://www.rgoarchitects.com/Files/fallacies.pdf.

Design patterns for microservices

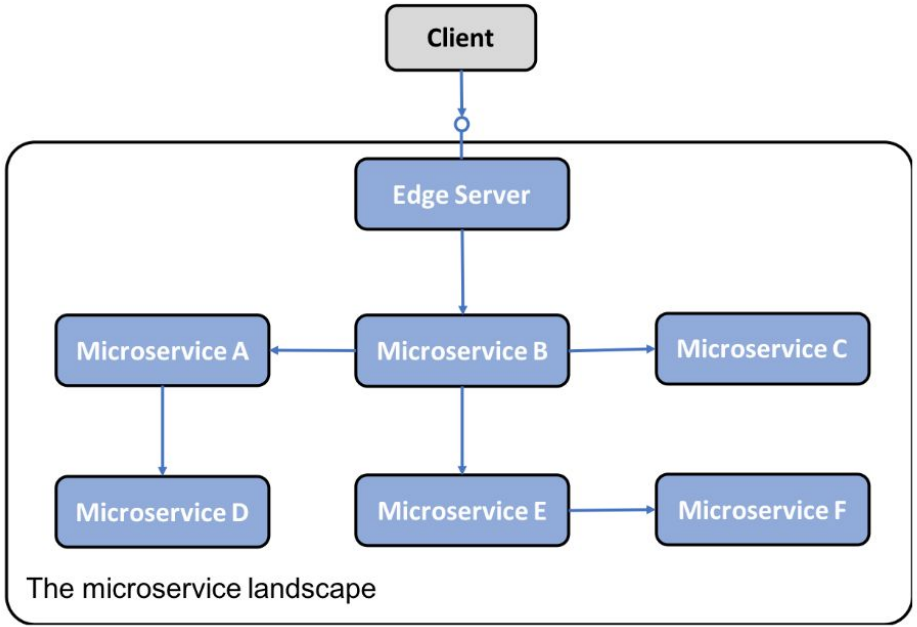
design patterns

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

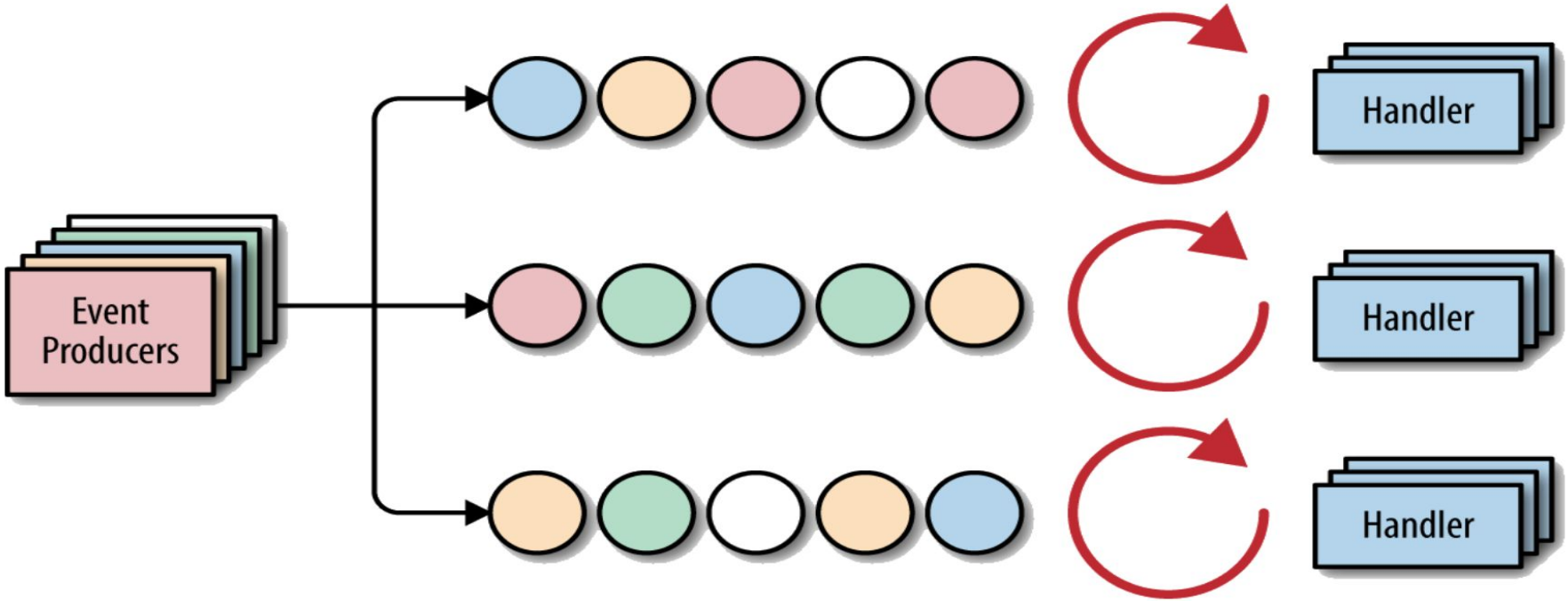
1. Service Discovery



2. Edge (Gateway/Load Balancer) Server



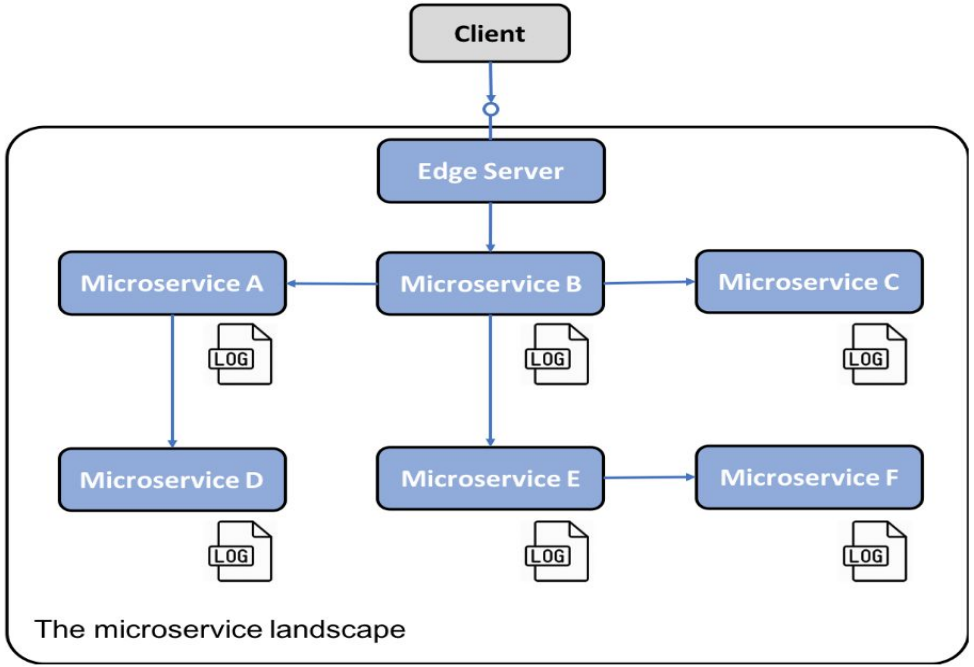
3. Reactive Microservices



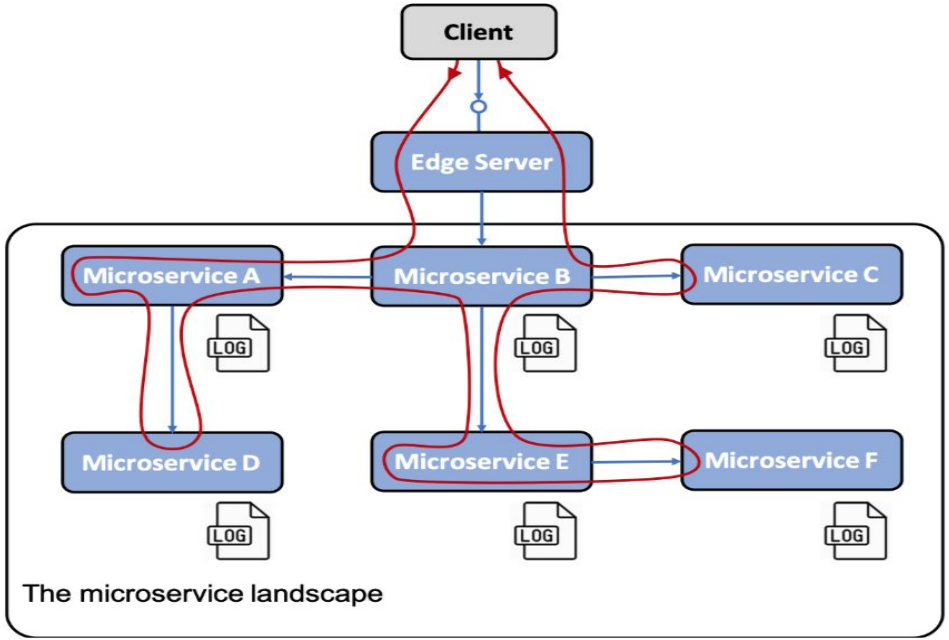
4. Central Configuration

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

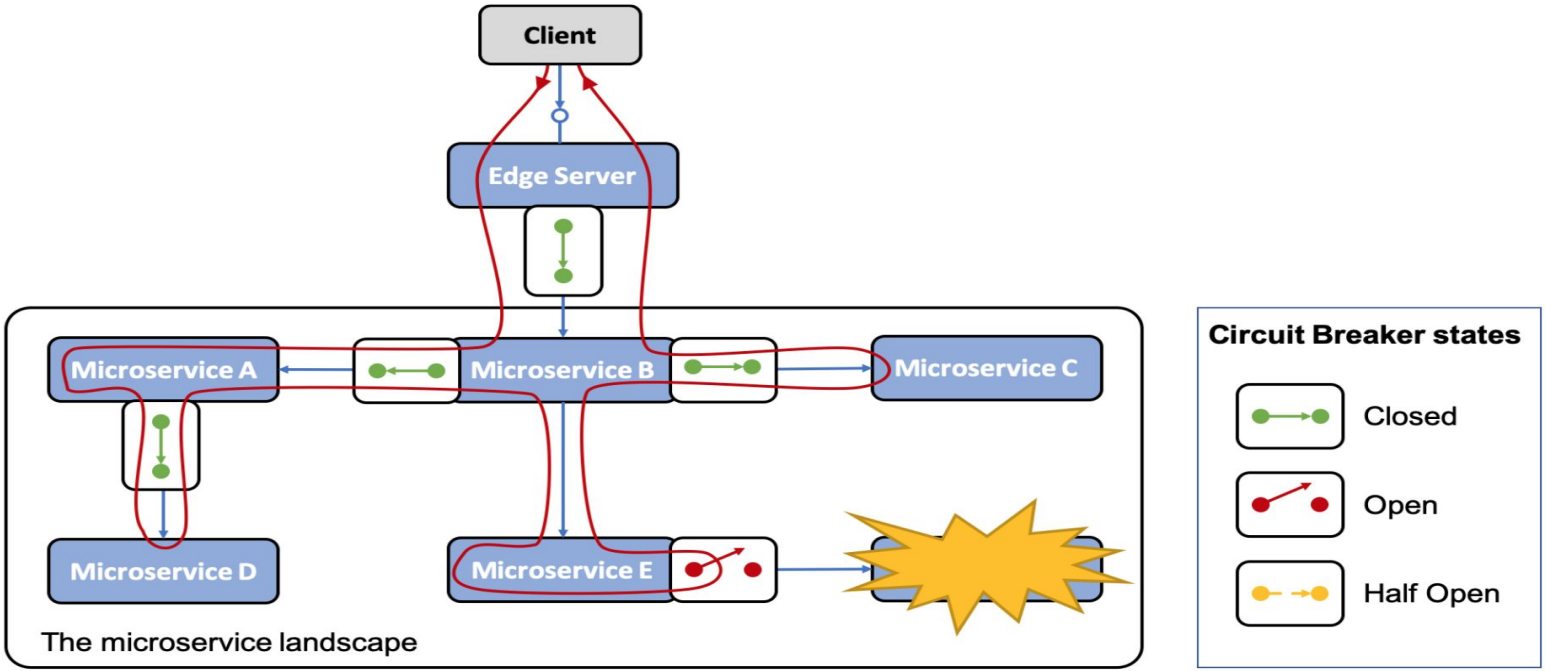
5. Centralized Log Analysis



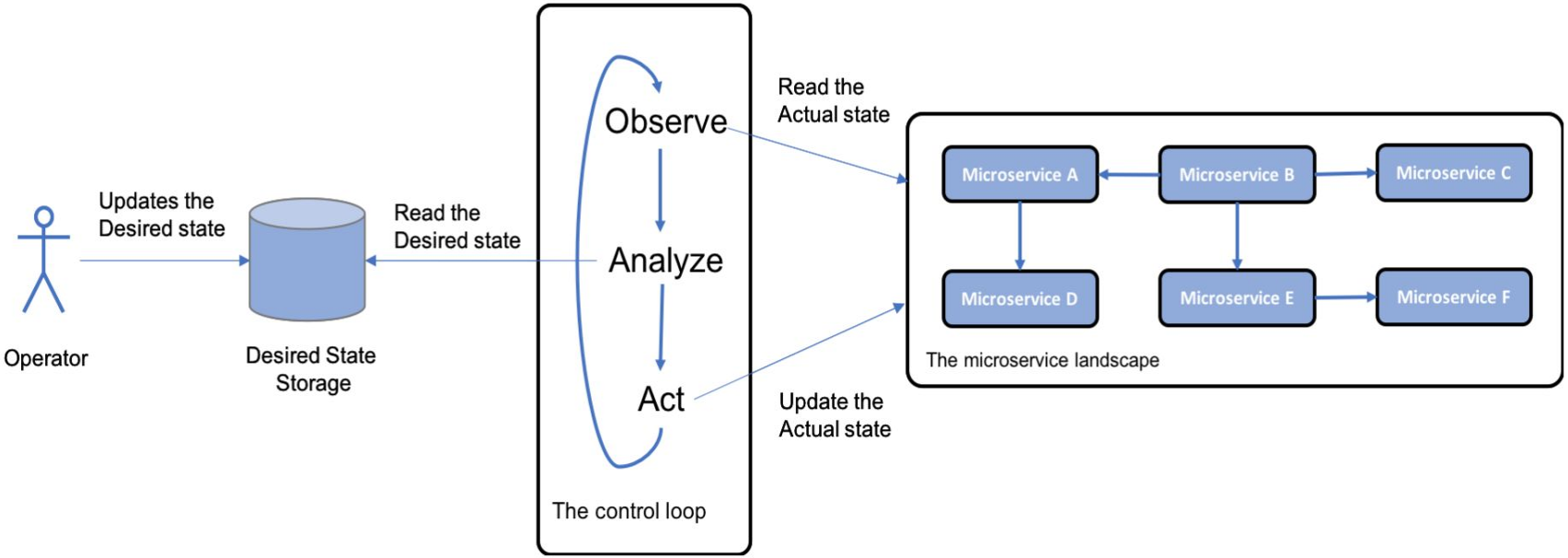
6. Distributed Tracing



7. Circuit Breaker



8. Control Loop - e.g k8s



9. Centralized monitoring & alarms