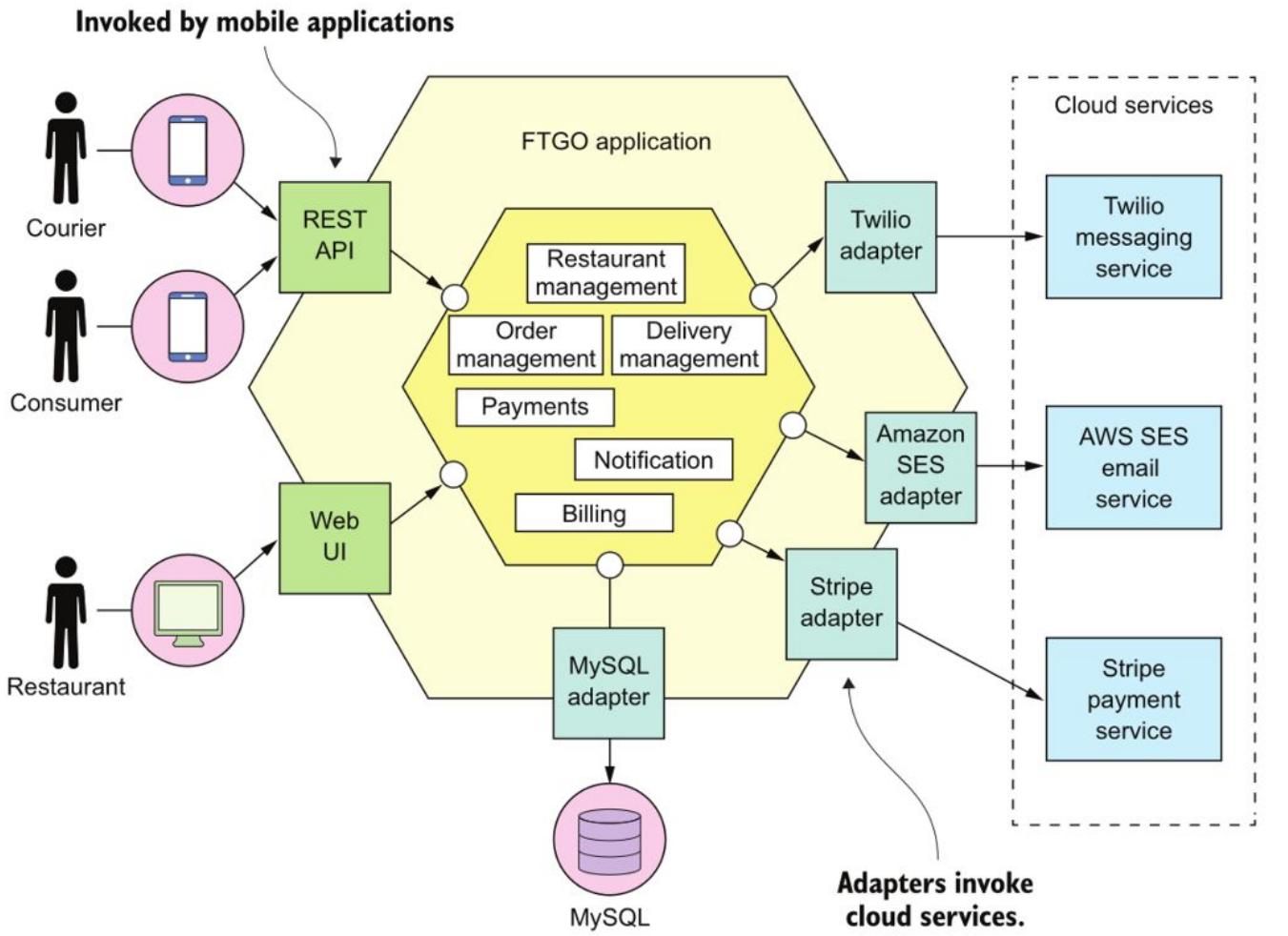


Event-Driven-Systems

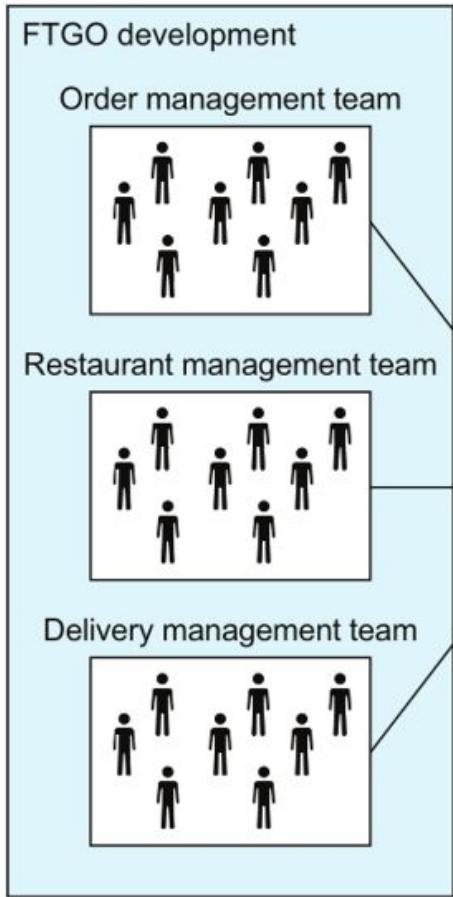
Evolving system will
require **change**

Blocker:
The Monolith Arch

FTGO application



Living in monolithic hell

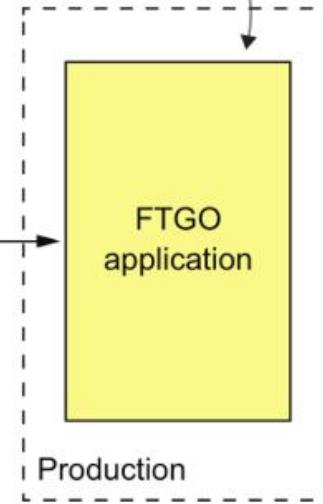
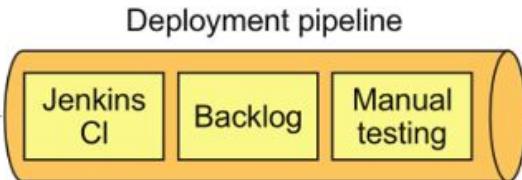


Large development organization

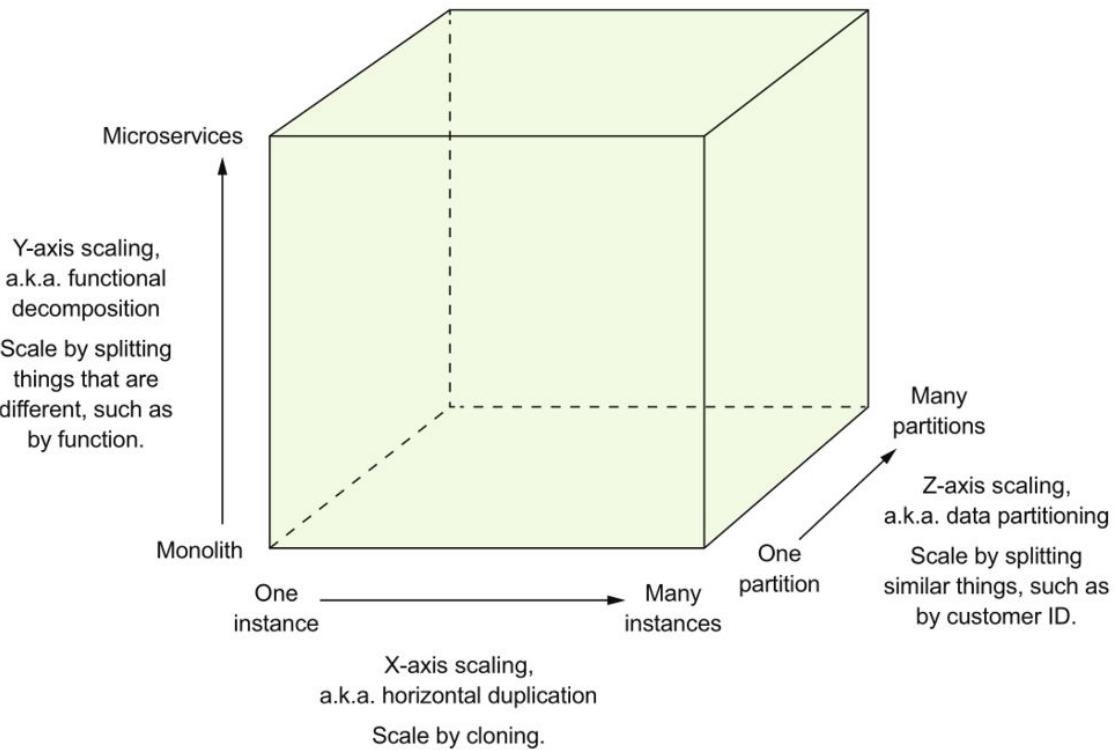
Single code base creates communication and coordination overhead.

Large, complex unreliable, difficult to maintain

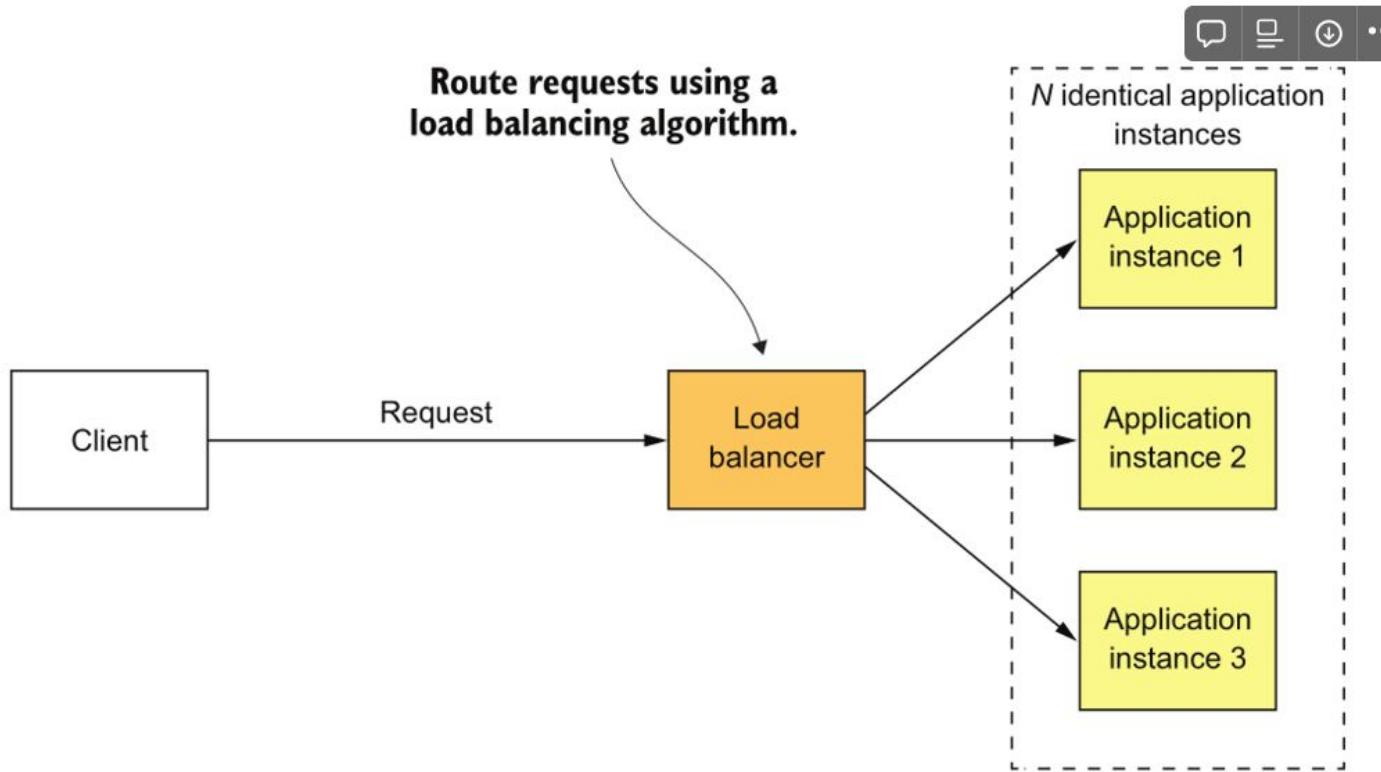
The path from code commit to production is arduous.
Changes sit in a queue until they can be manually tested.



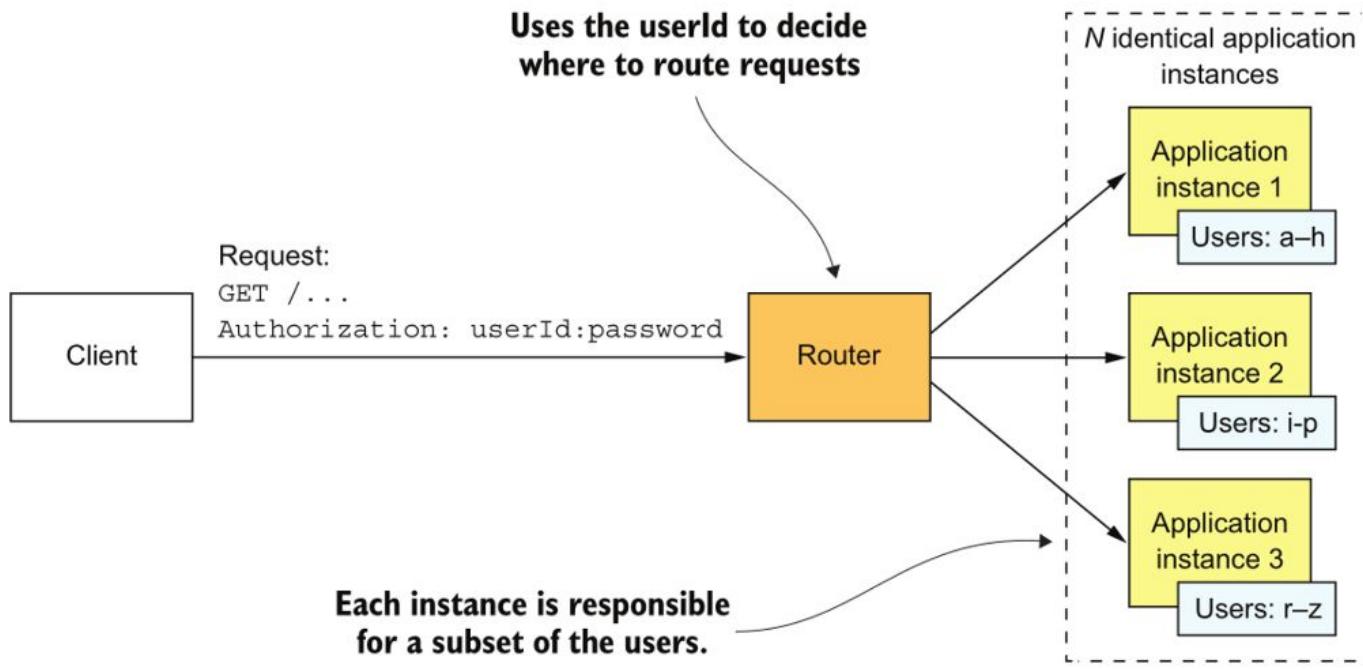
Scale cube



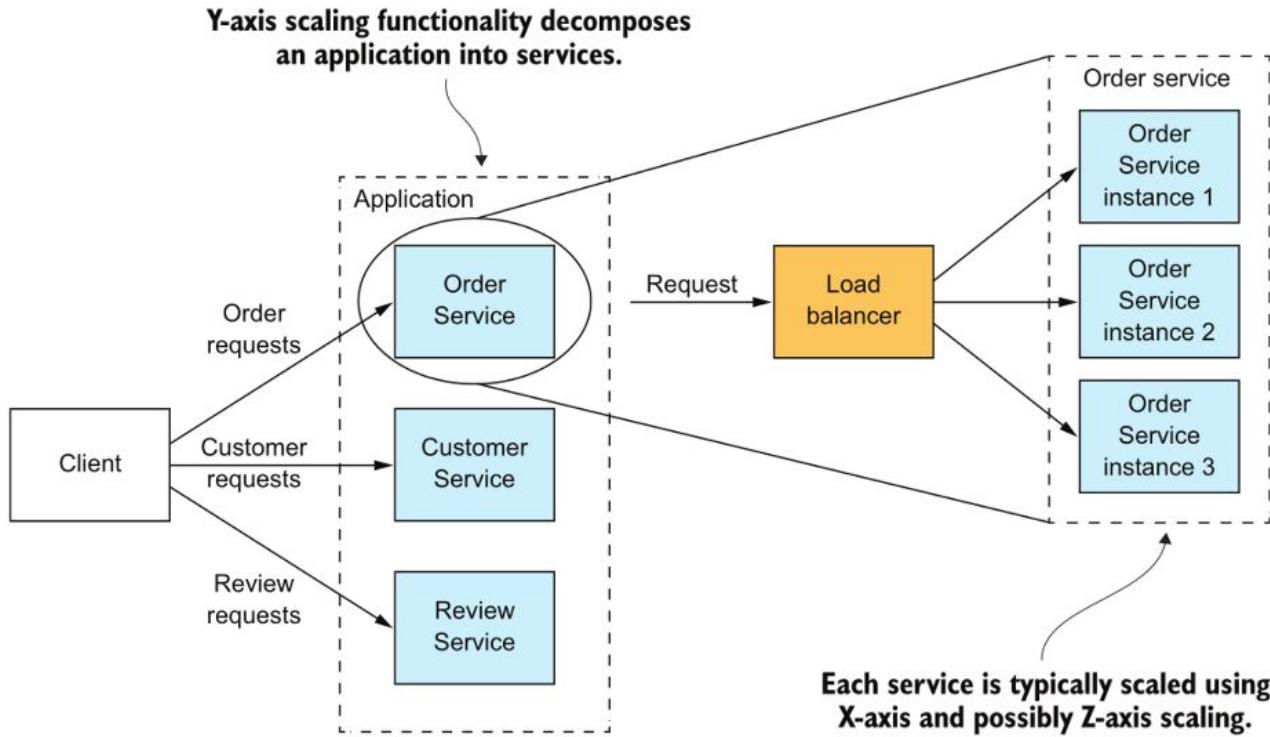
X-axis scaling



Z-axis scaling



Y-axis scaling



Solution

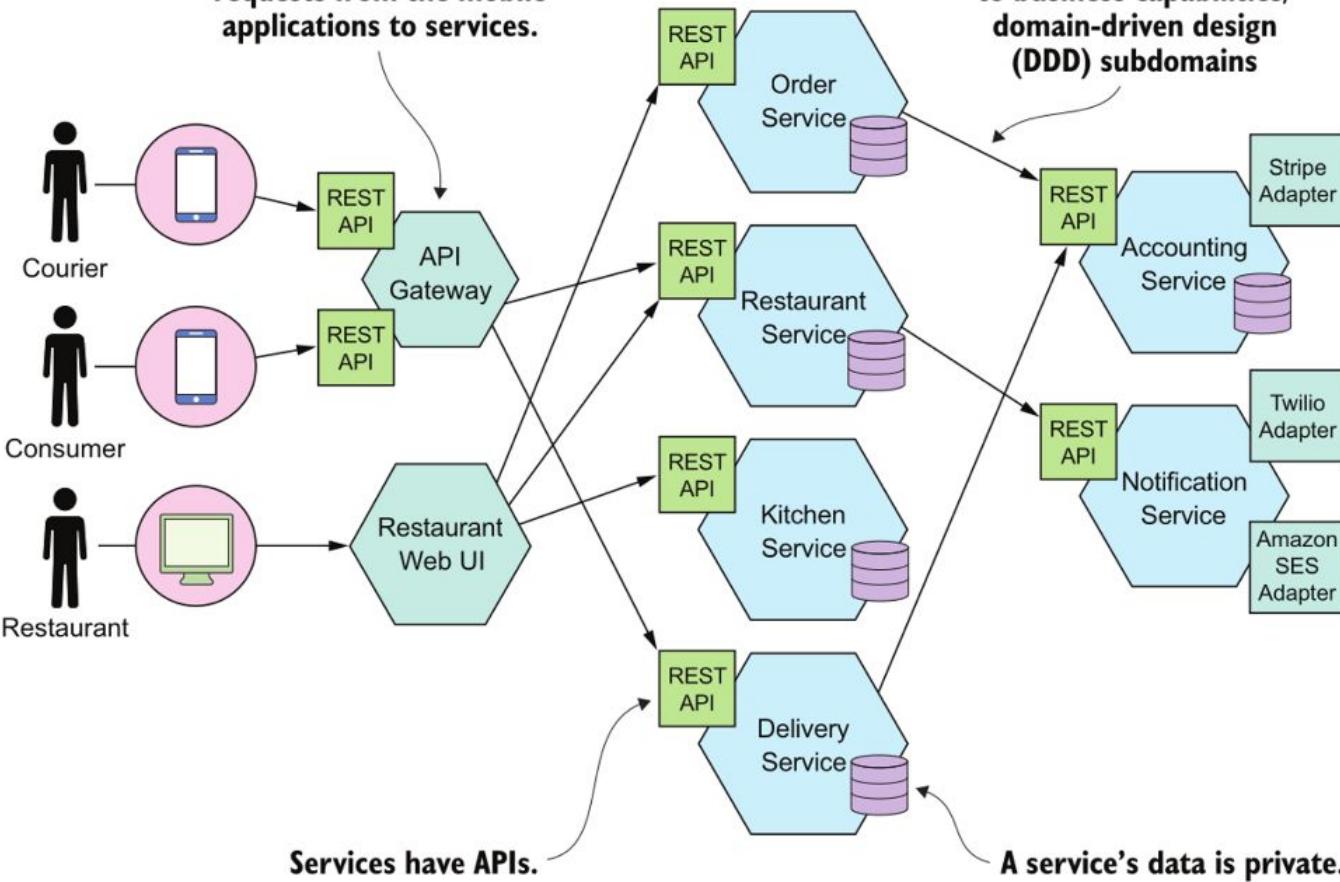
Microservices

Or

Distributed systems

FTGO application

The API Gateway routes requests from the mobile applications to services.

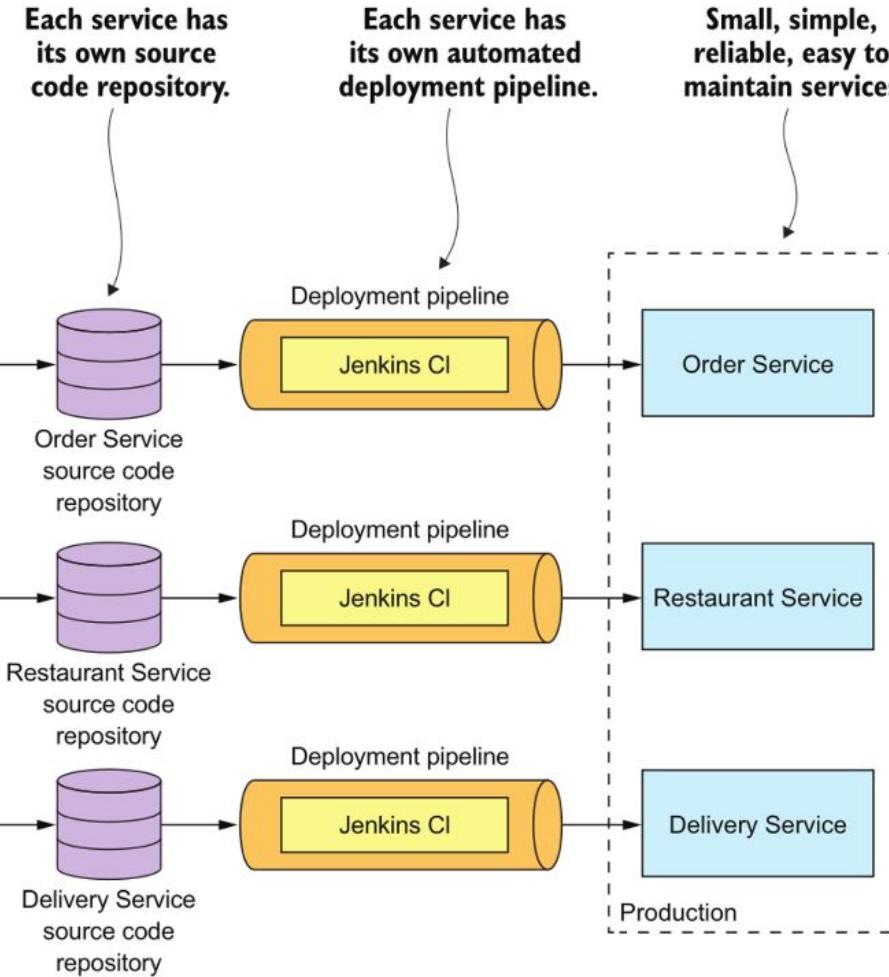
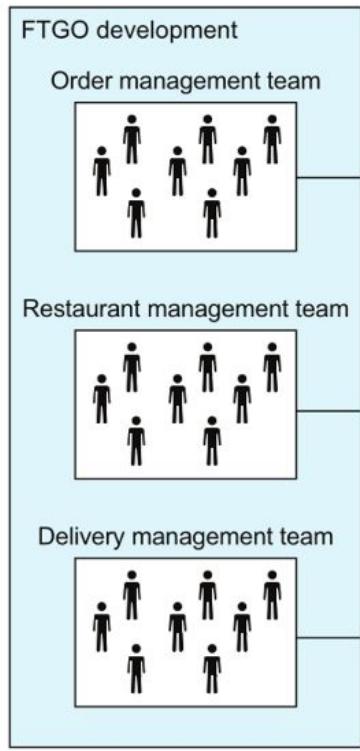


**Small, autonomous,
loosely coupled teams**

**Each service has
its own source
code repository.**

**Each service has
its own automated
deployment pipeline.**

**Small, simple,
reliable, easy to
maintain services**



Benefits

- 1- Services are small & easily maintainable
- 2- Services are independently deployable & scalable.
- 3- Enables teams to be autonomous.
- 4- It has better fault isolation.
- 5- It allows easy experimenting & adoption of new technologies

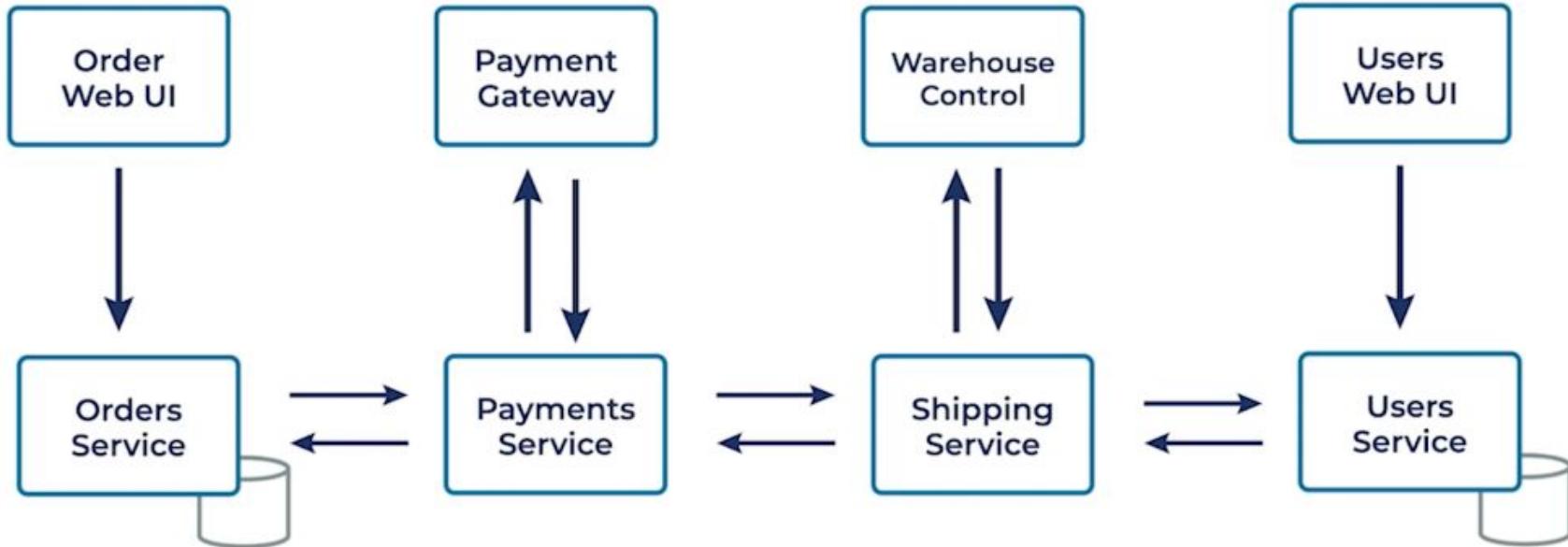
Types of Microservices

By communication

1. RPC (Request & Response)
2. Event-Driven (Via Messages)

I'm here, for Any Q

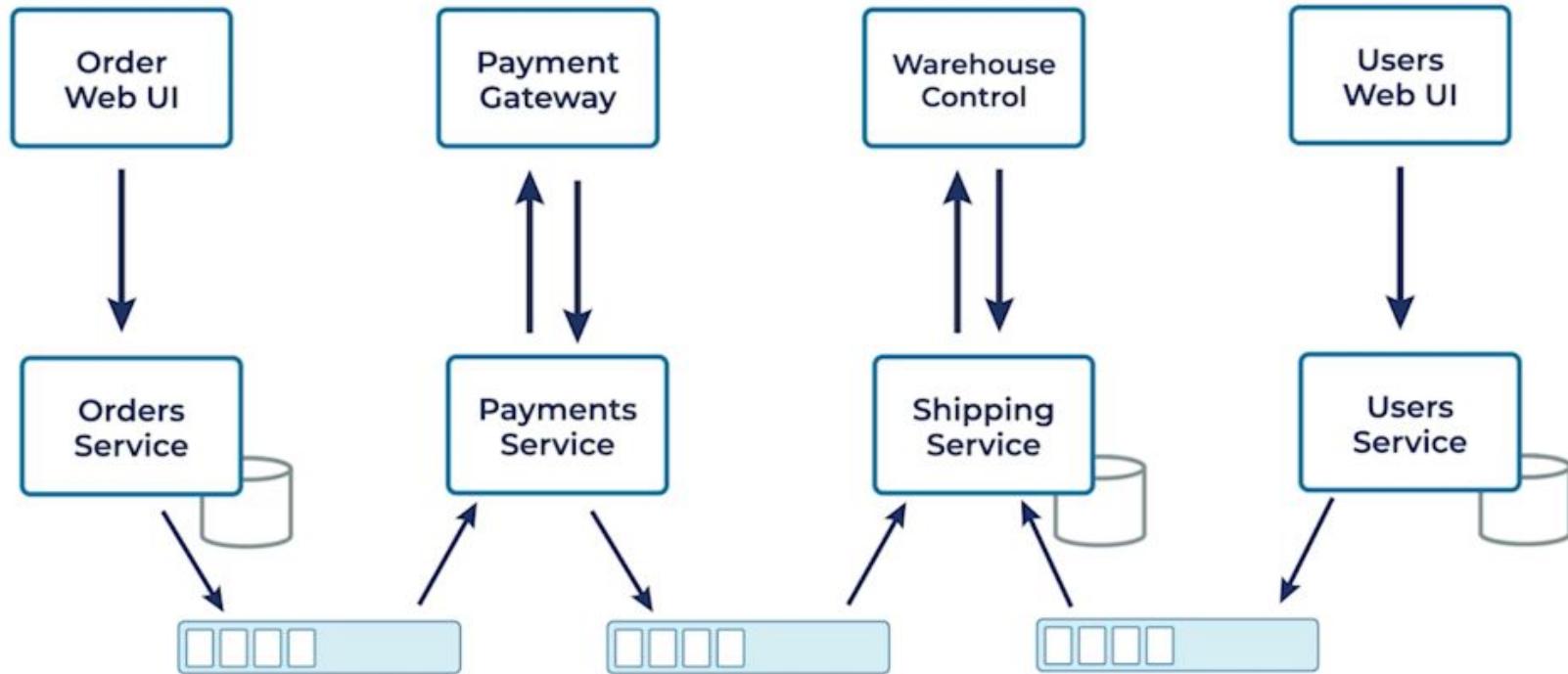
RPC based Microservices



RPC - challenges

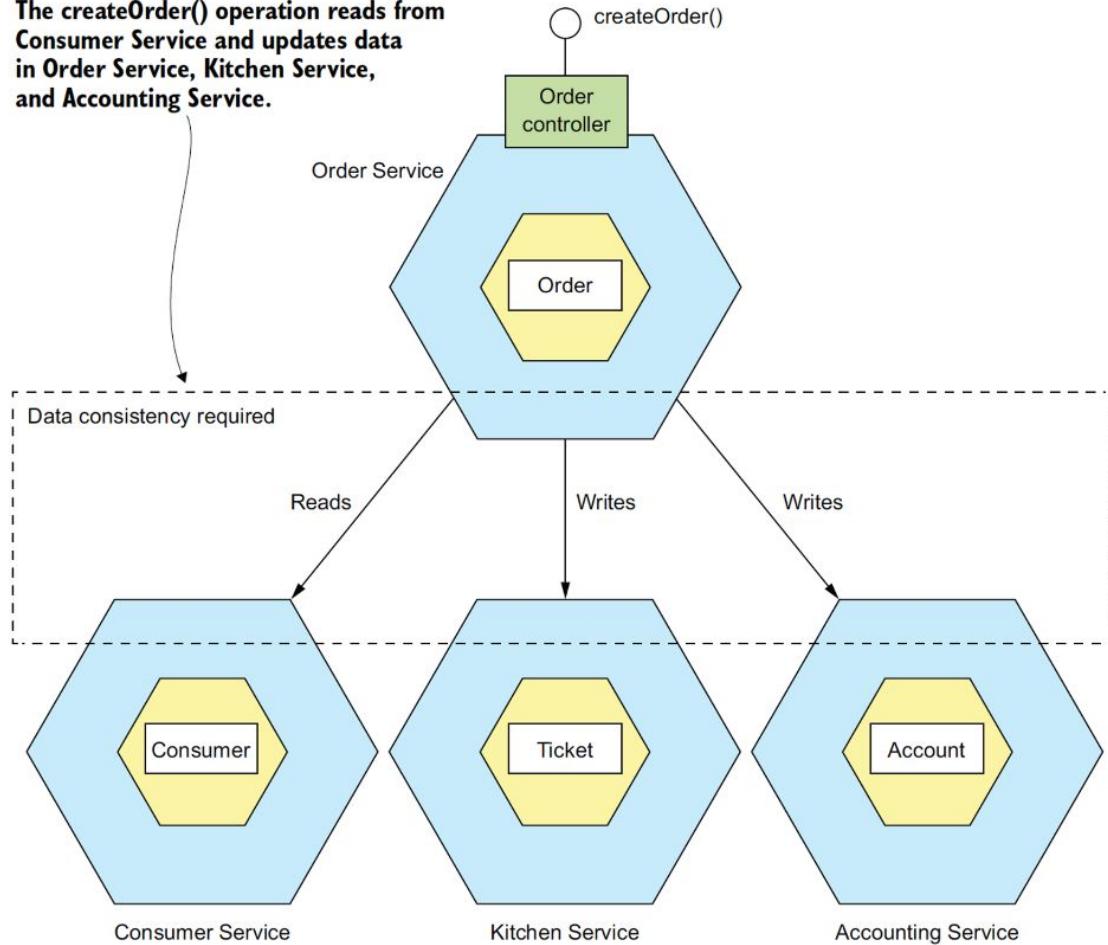
1. Cross language communication
2. Service-discovery
3. Load-balancing
4. Network-latency
5. Reliability
6. Versioning & Compatibility

Microservices (Refactoring to **Events**)

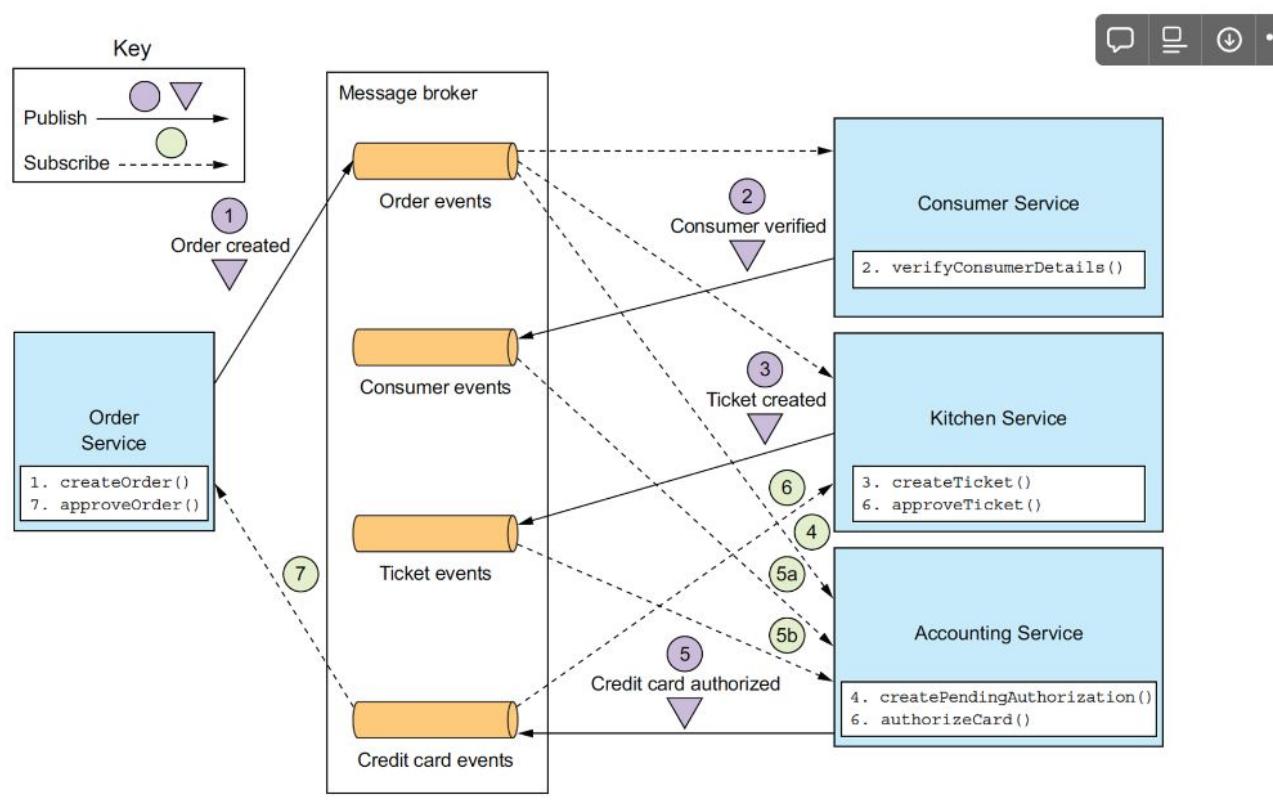


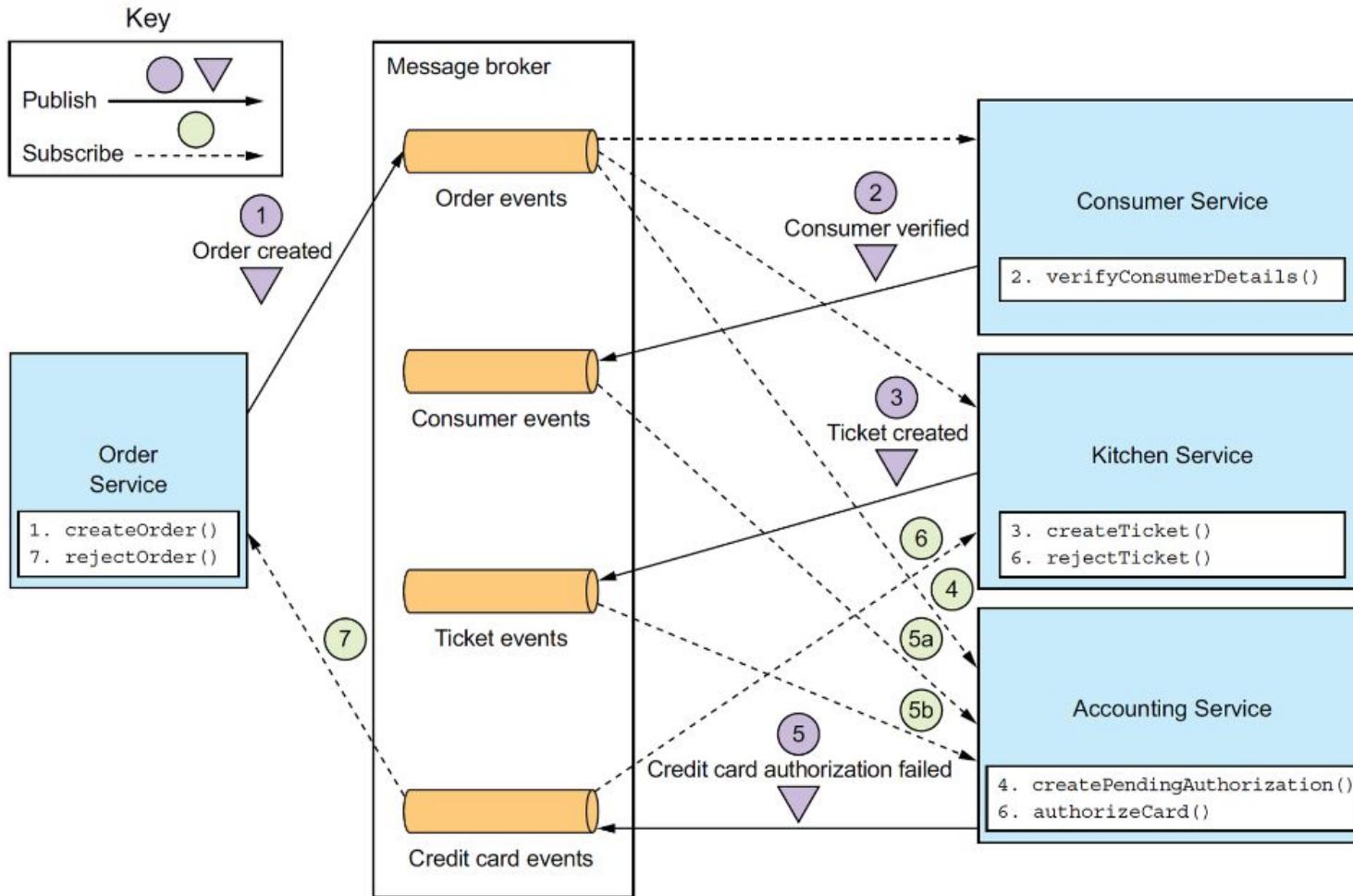
Example

The `createOrder()` operation reads from Consumer Service and updates data in Order Service, Kitchen Service, and Accounting Service.



Event-Driven Microservices

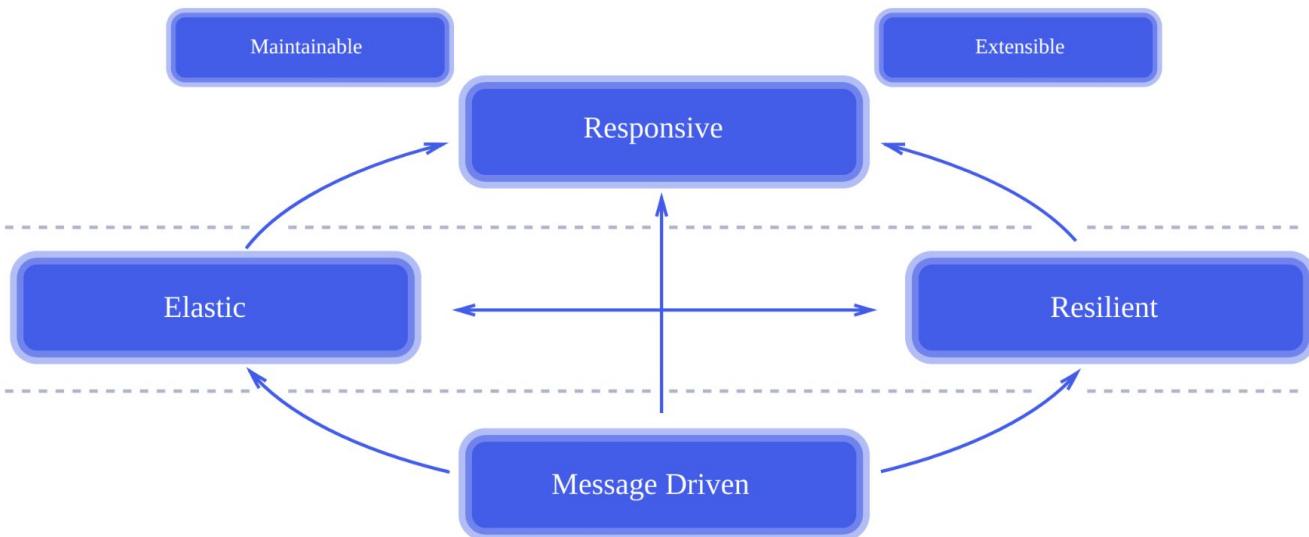




Event-Driven - Advantages

1. Decoupling of services
2. Scalability
3. Resilience and Fault Tolerance
4. Improved Responsiveness & Performance
5. Ease of Integration
6. Enables Reactive Programming Models

The Reactive Manifesto



<https://www.reactivemanifesto.org/>

How to send/store/react(receive)the event(s) ?

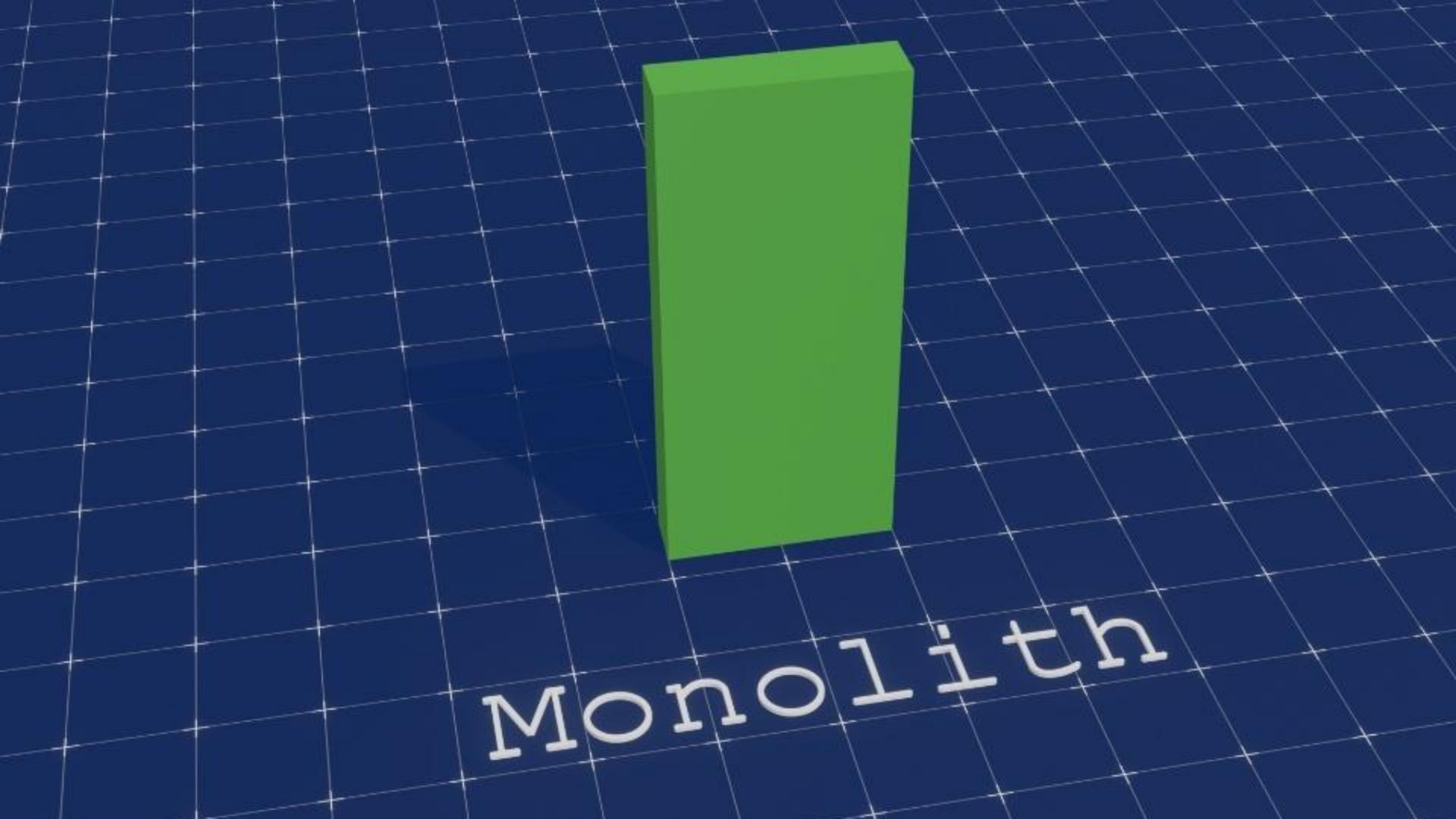
Using **Messaging Systems**,

Ex.

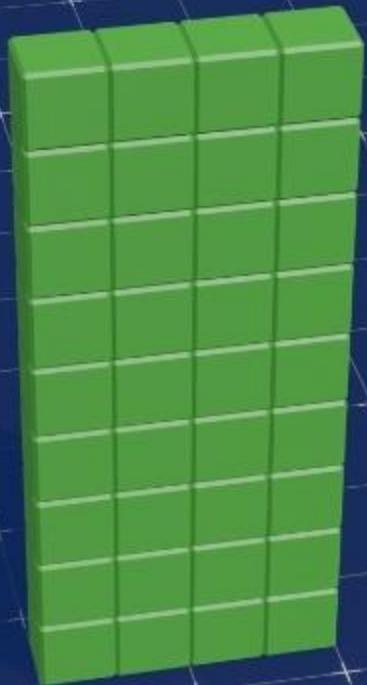
1. **Apache Kafka**
2. RabbitMQ, ActiveMQ, IBM MQ,..
3. Microsoft Azure Event Hub
4. Google cloud Pub/Sub

.....

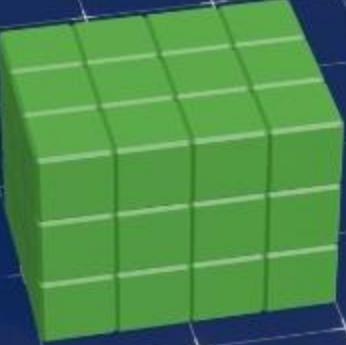
Visual Summary

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

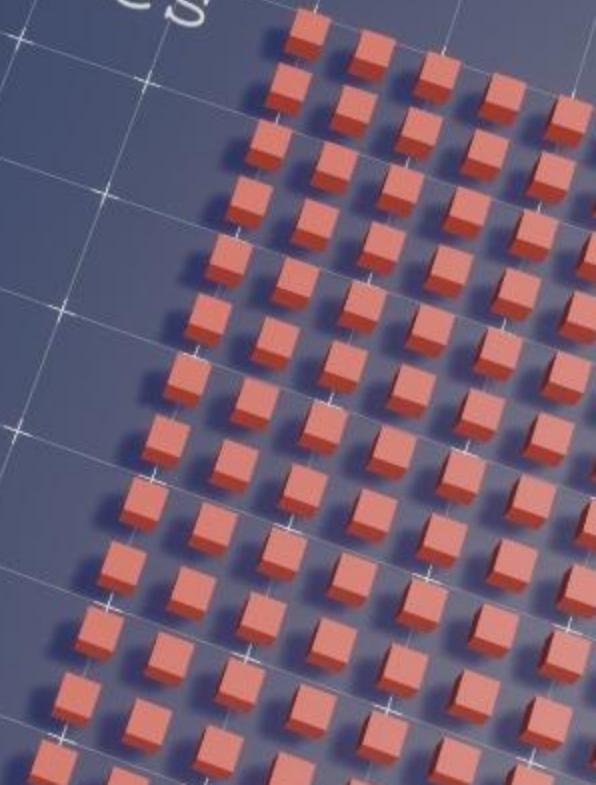
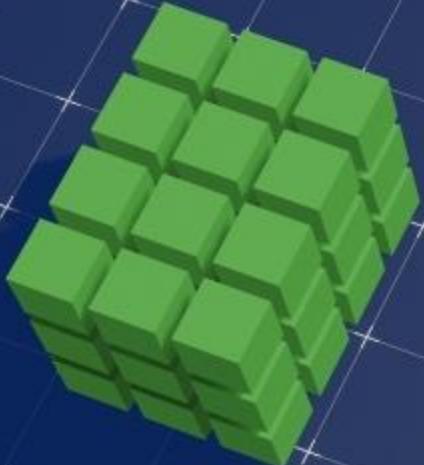


Monolith
Modules

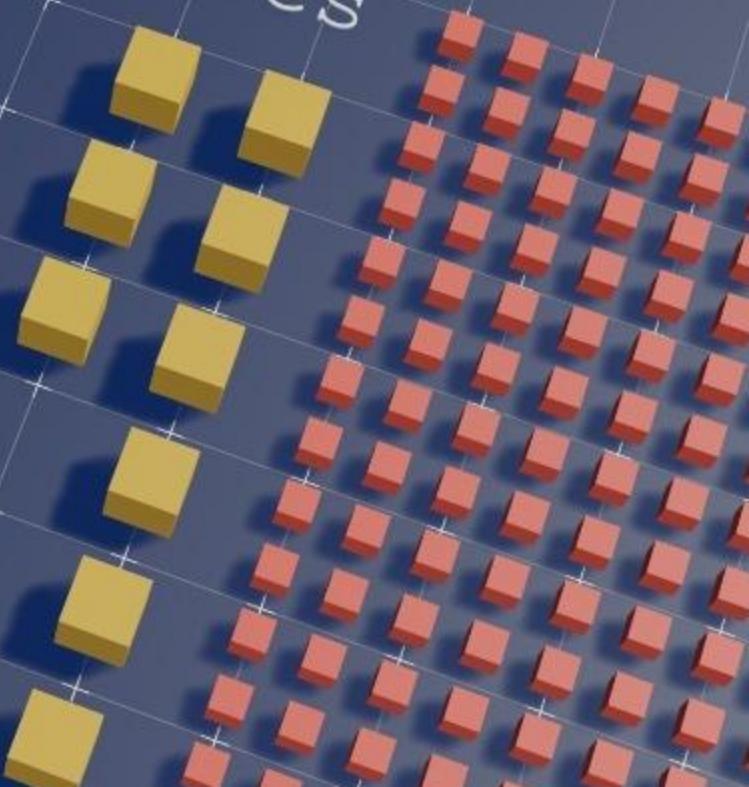
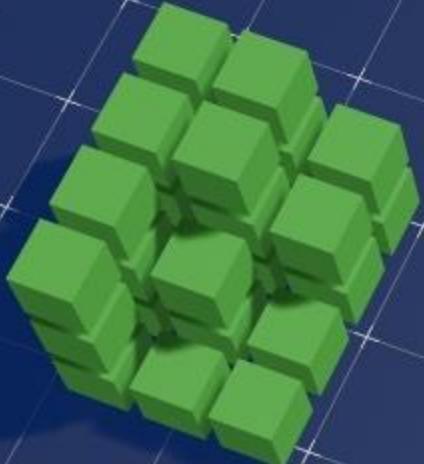


Monolith
Modules

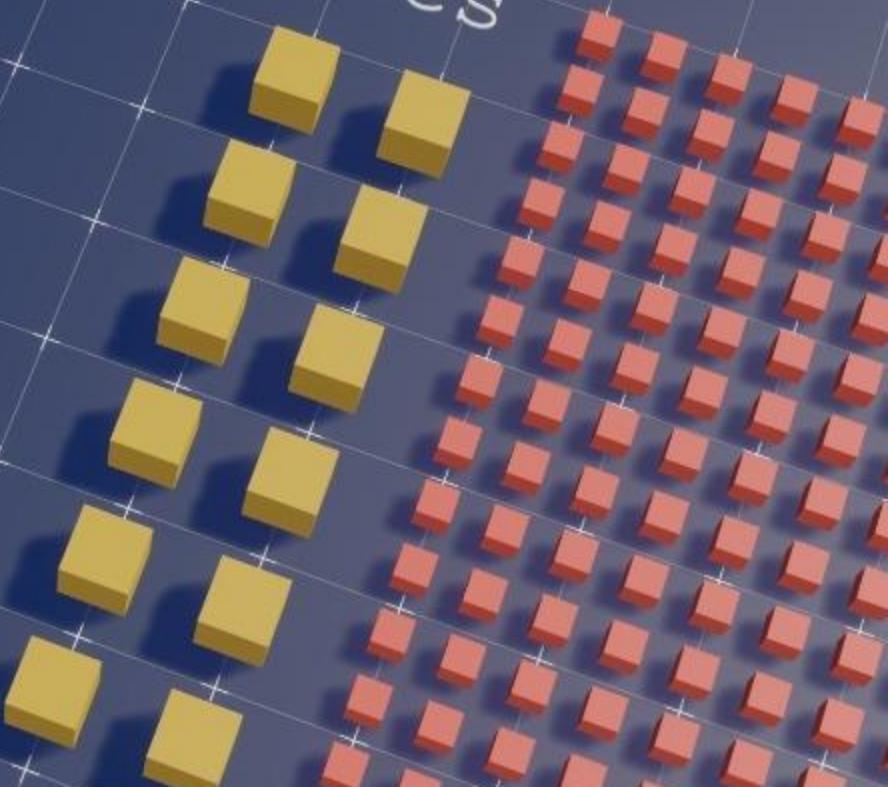
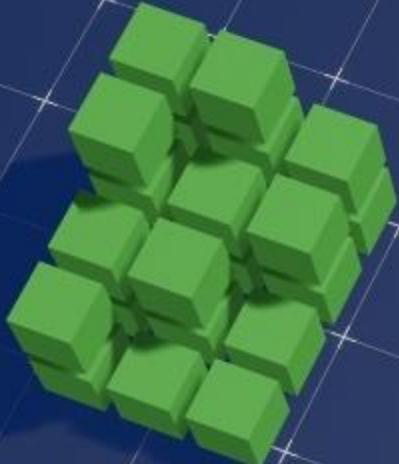
Microservices



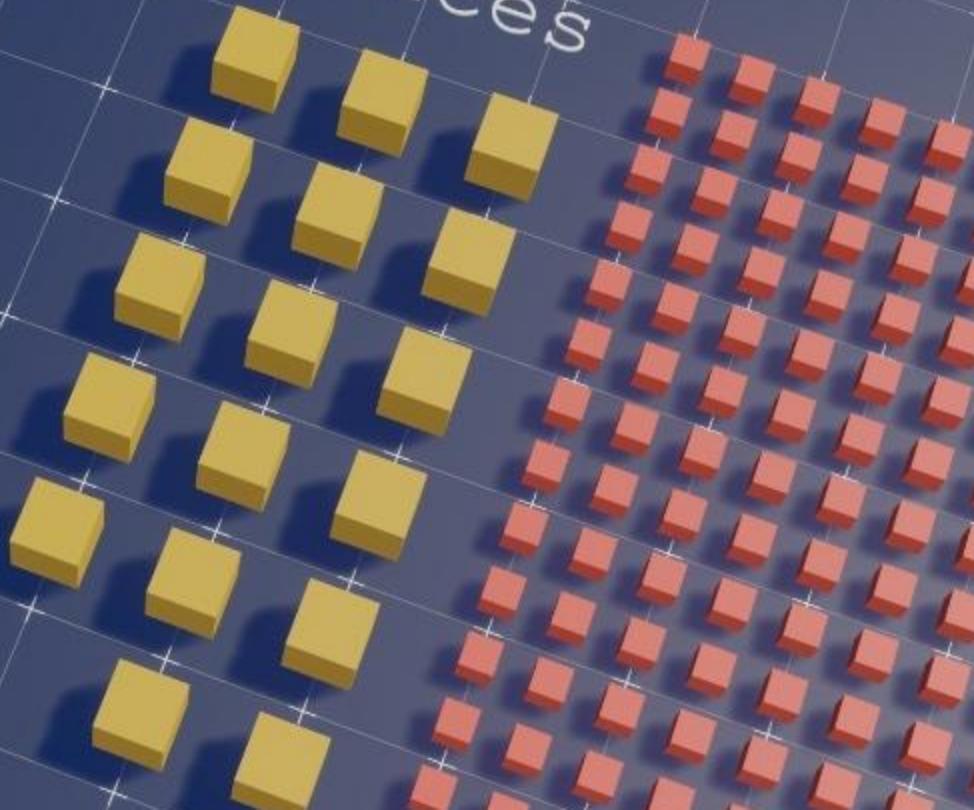
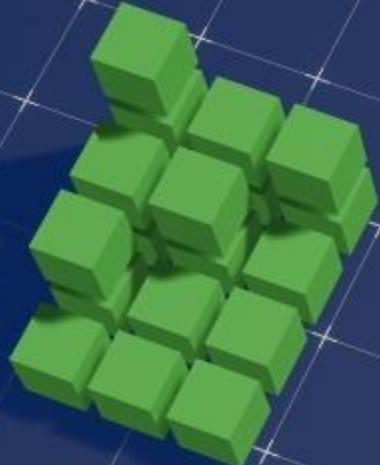
Microservices



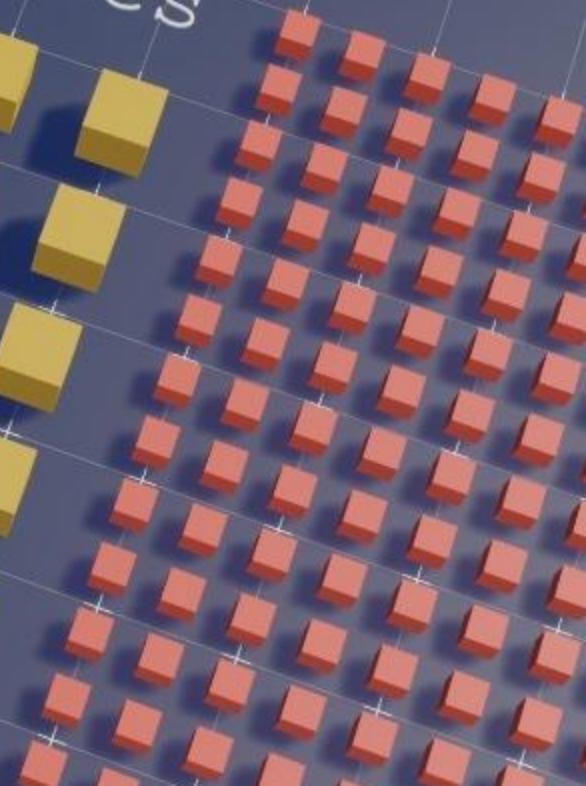
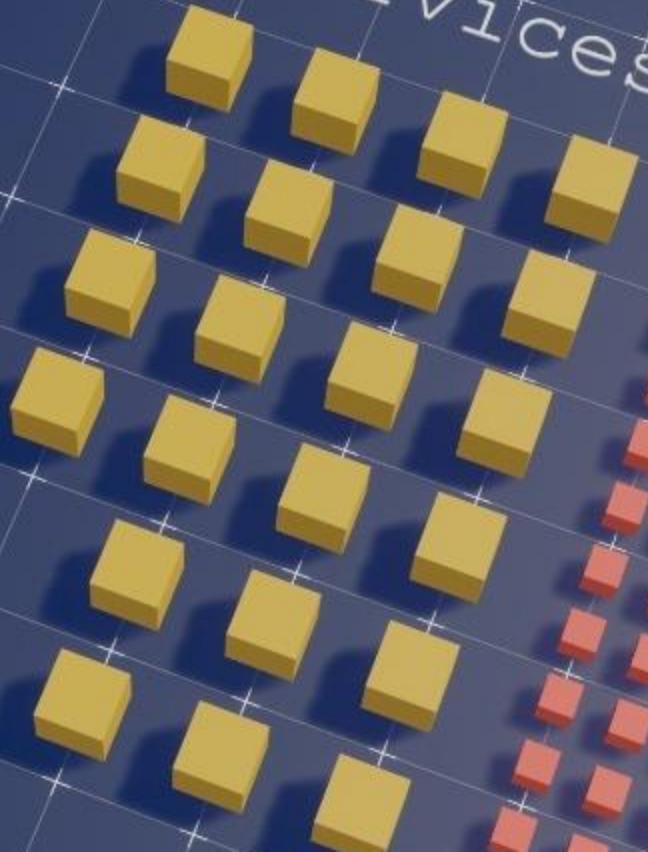
Microservices



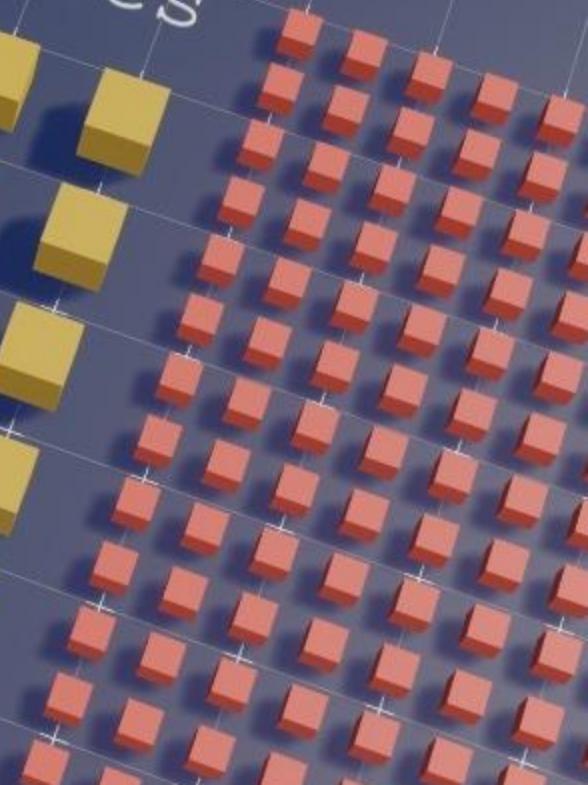
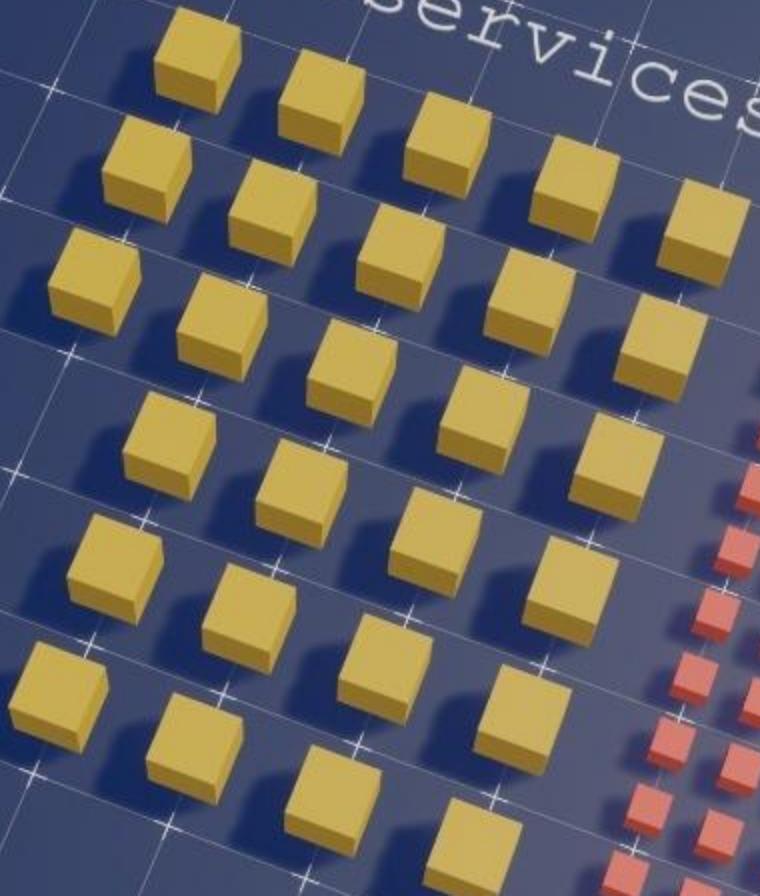
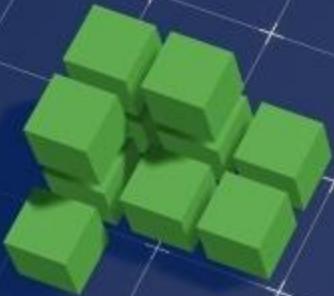
Microservices



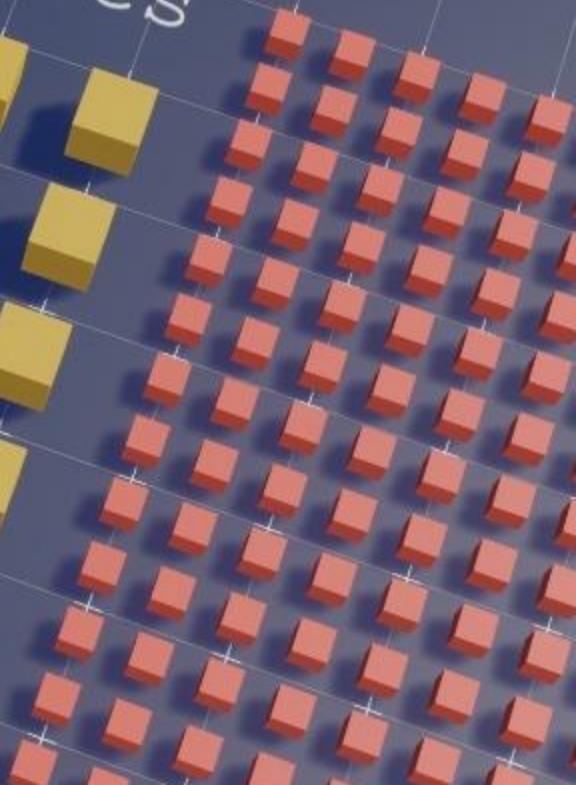
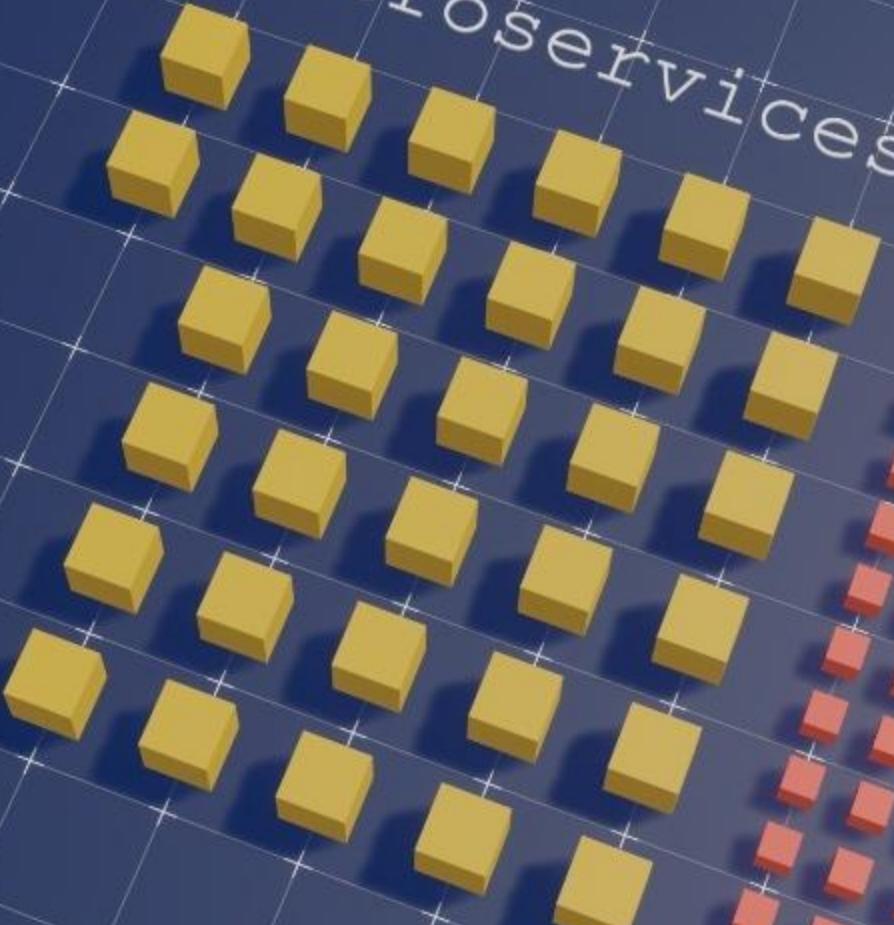
Microservices



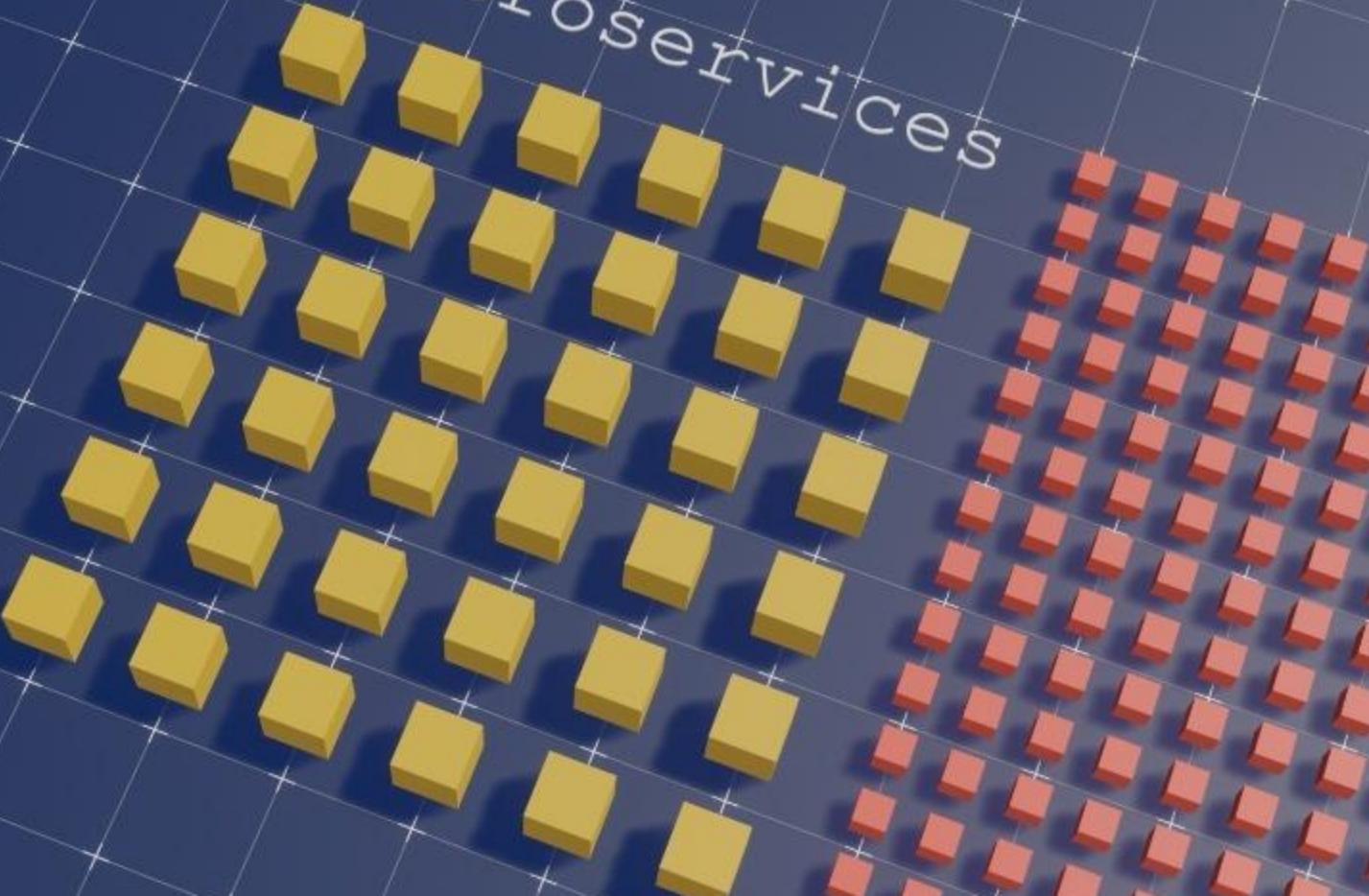
Microservices

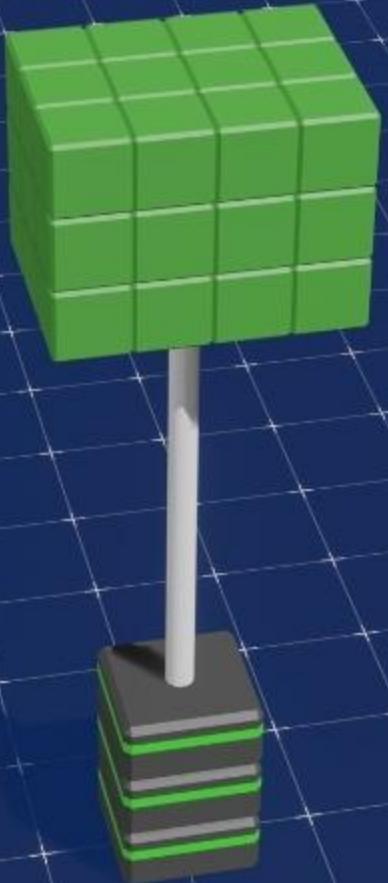


Microservices

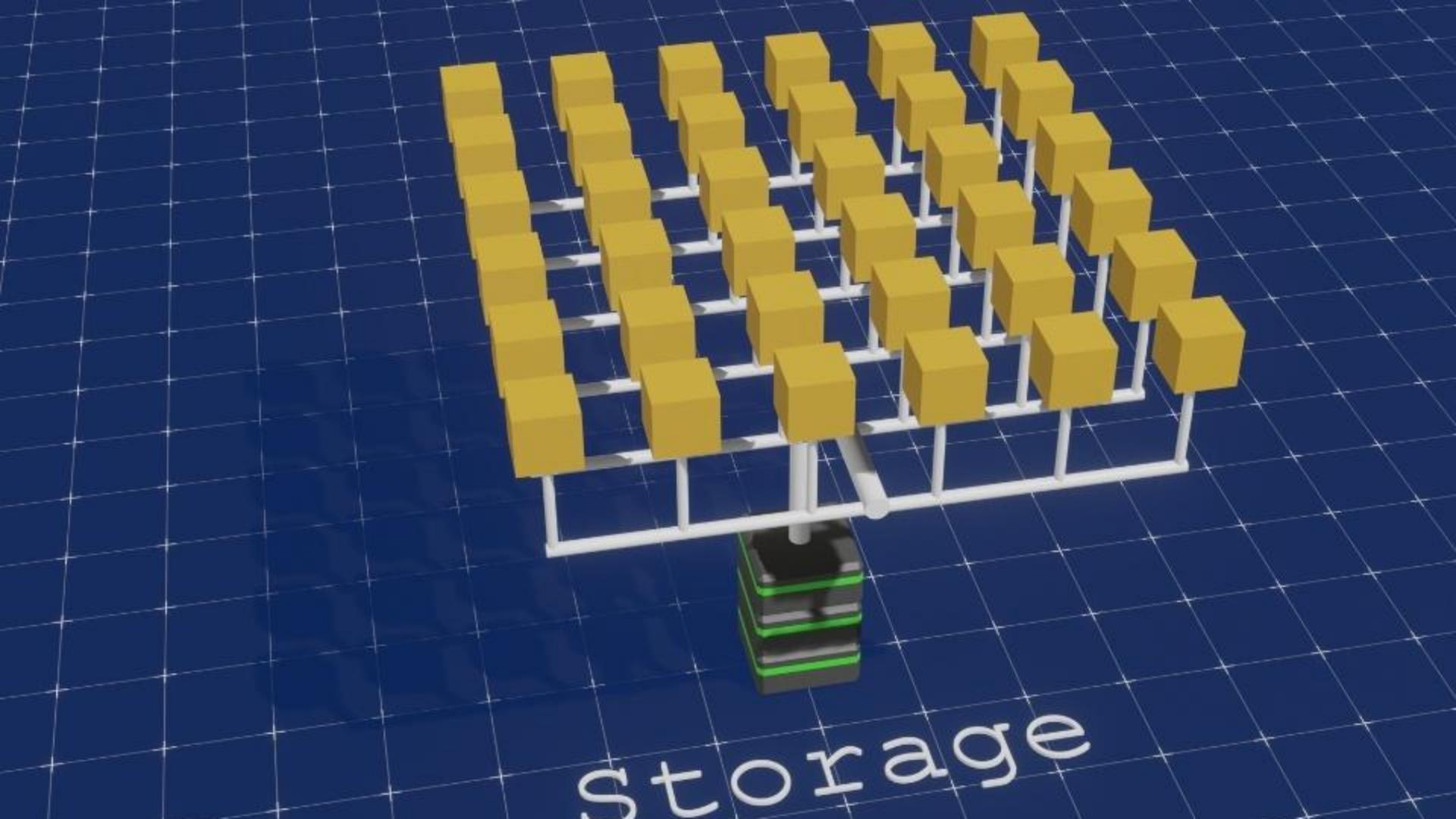


Microservices

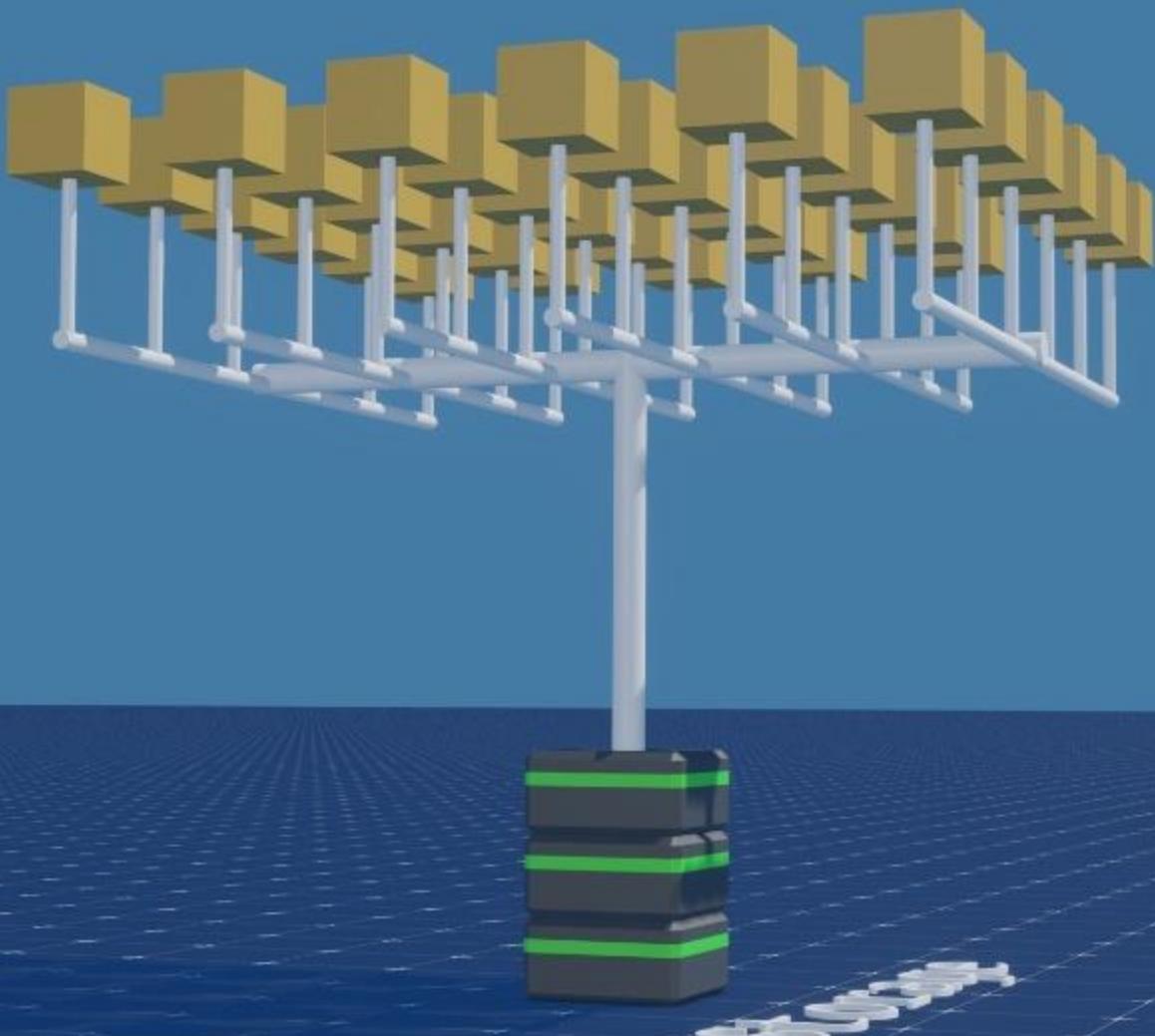


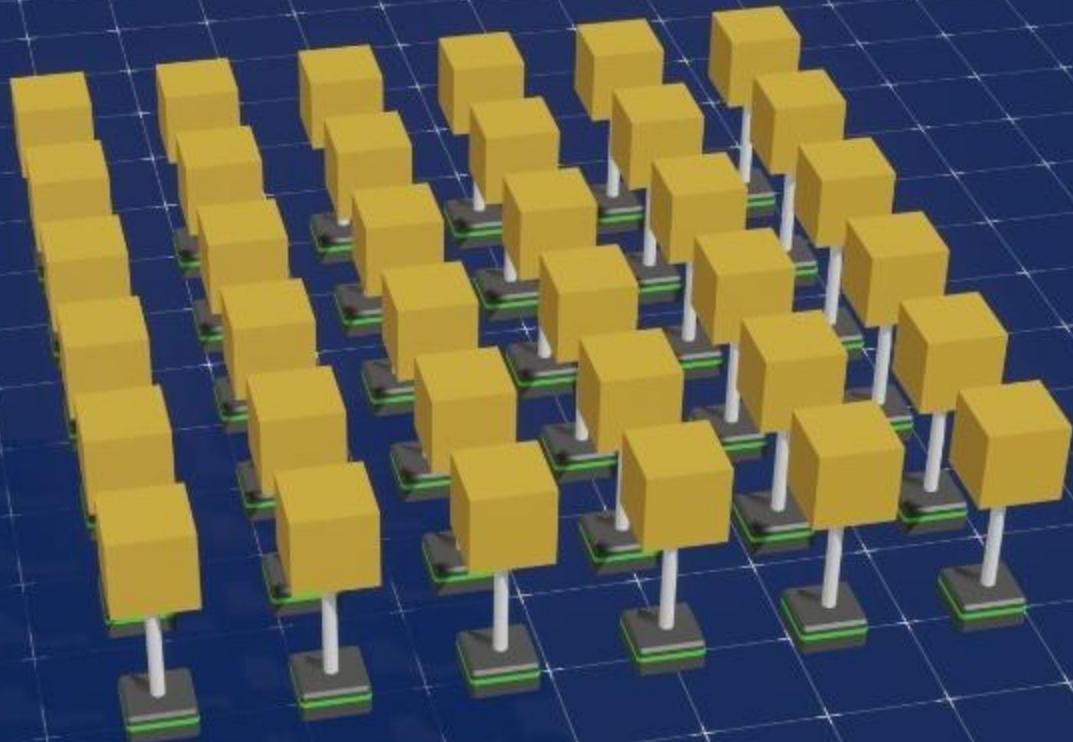


Storage

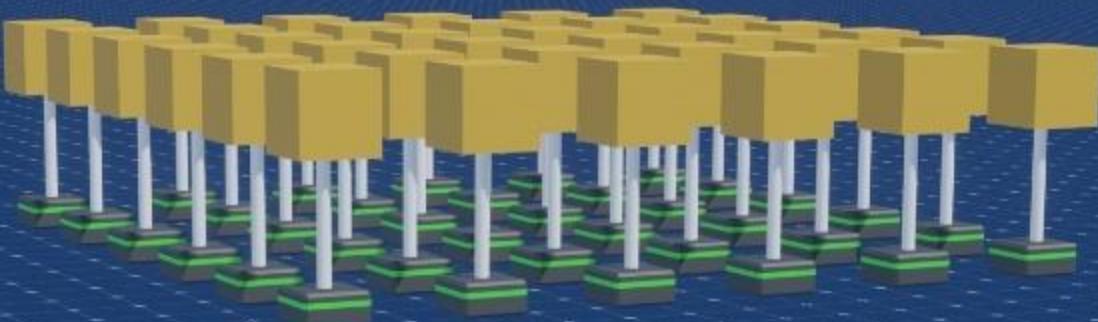


Storage

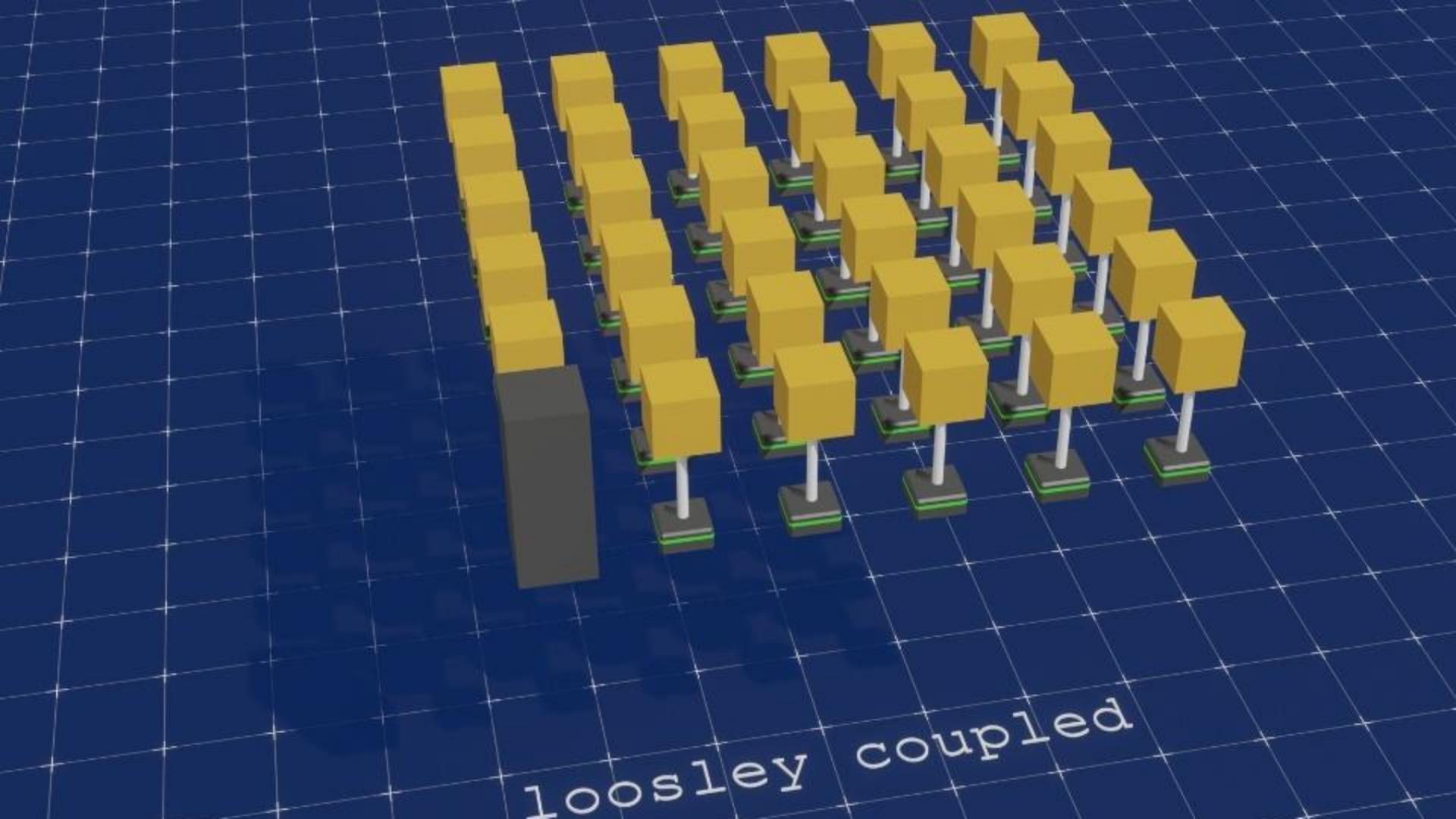




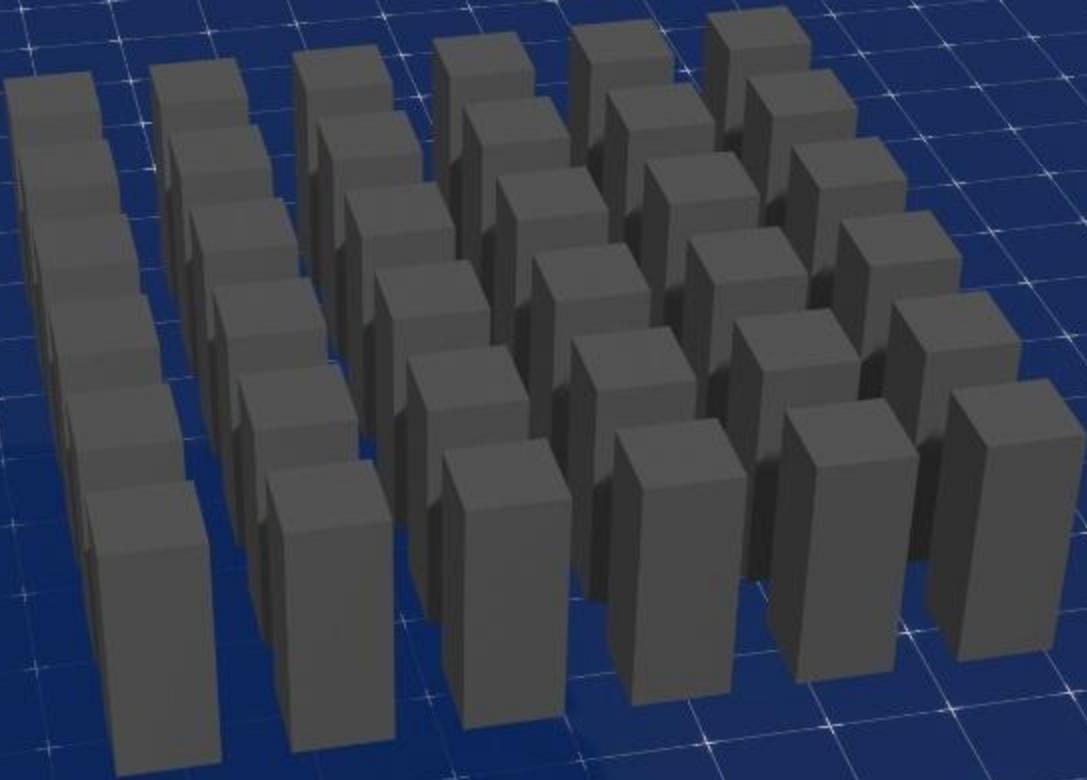
loosley coupled



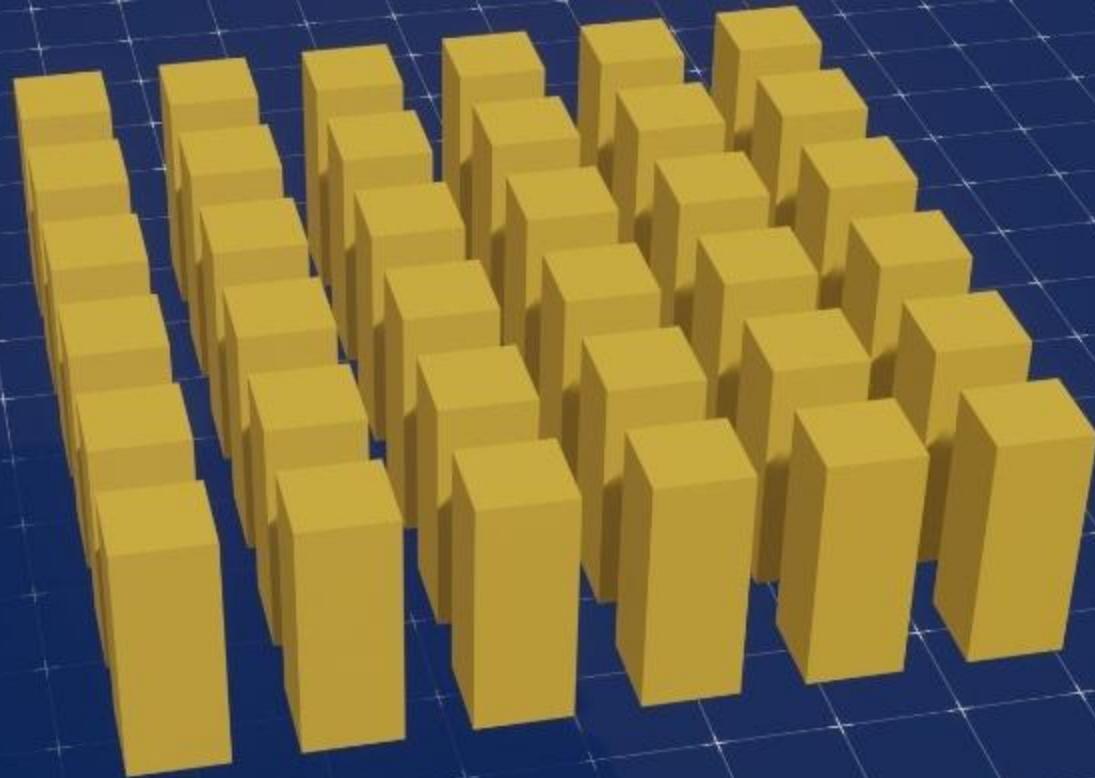
loosely coupled



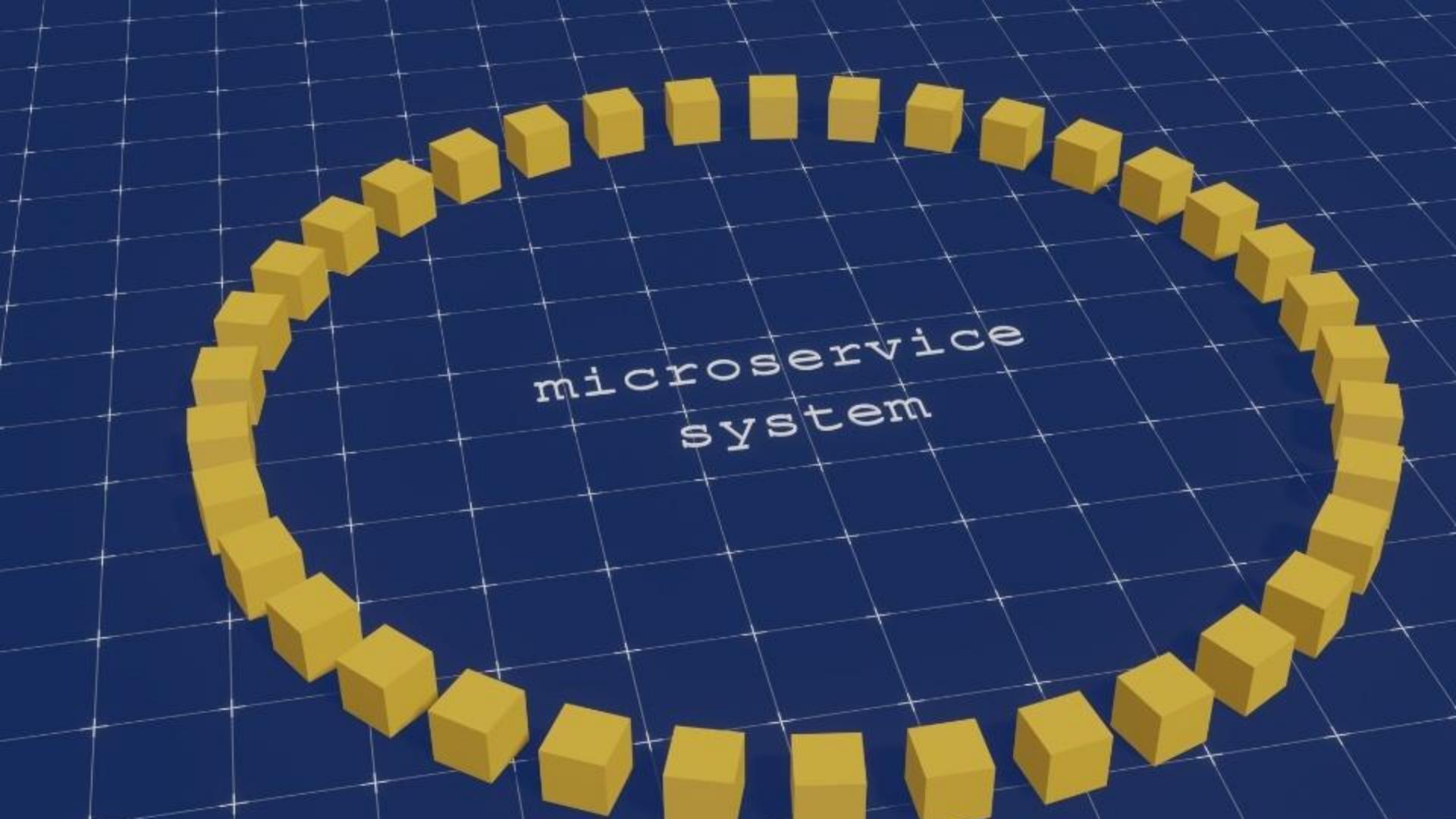
loosley coupled



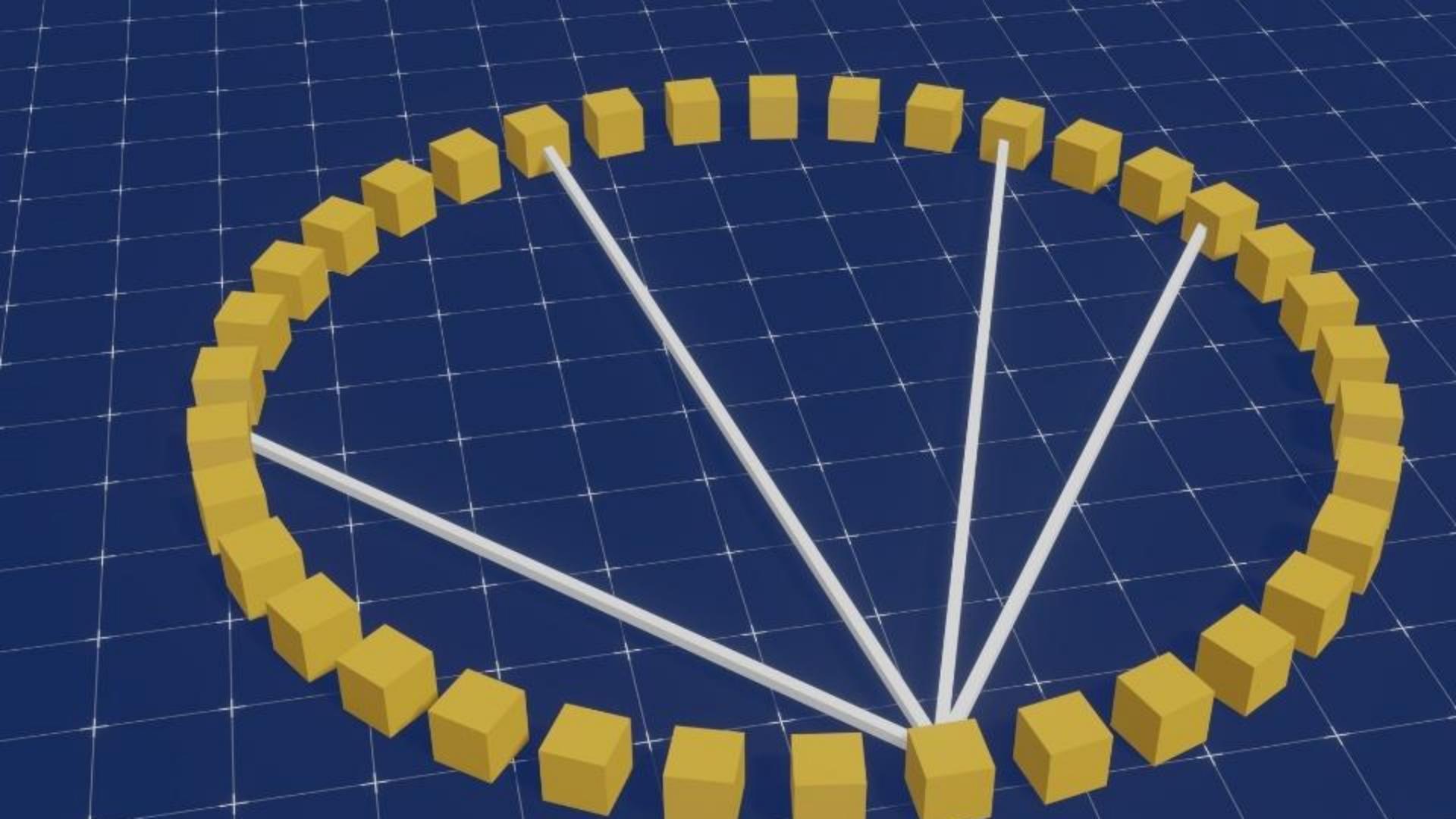
loosley coupled

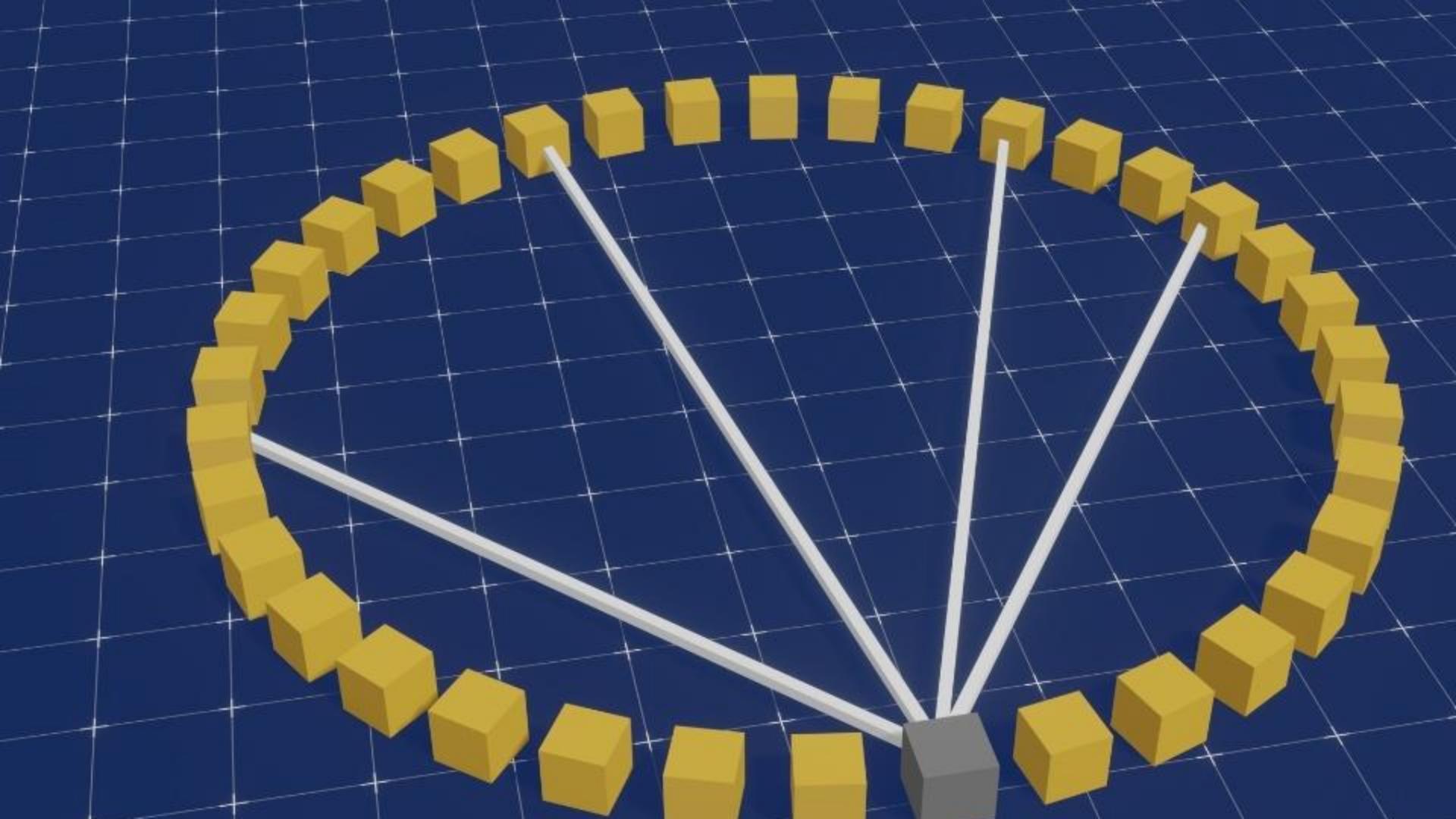


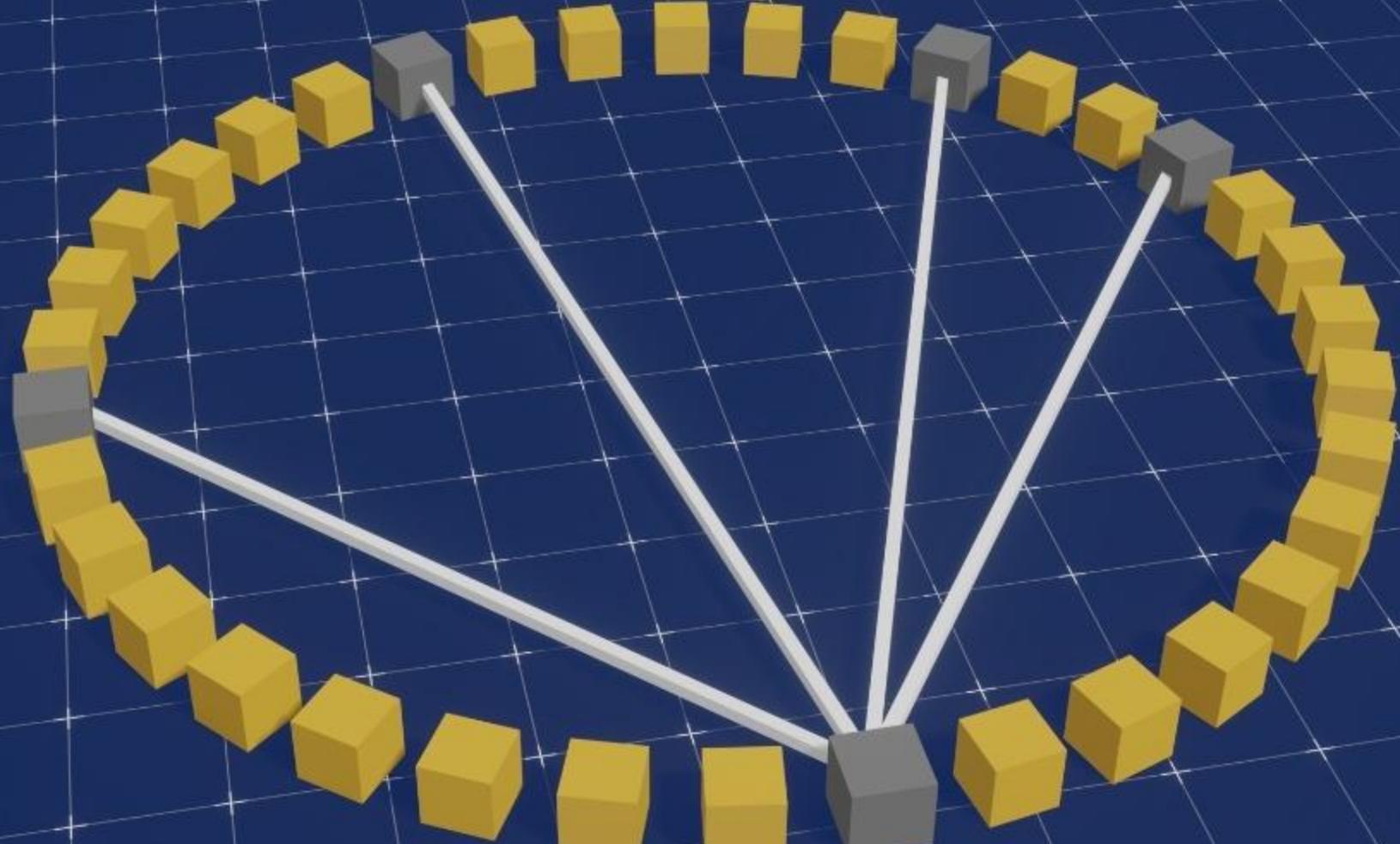
loosley coupled



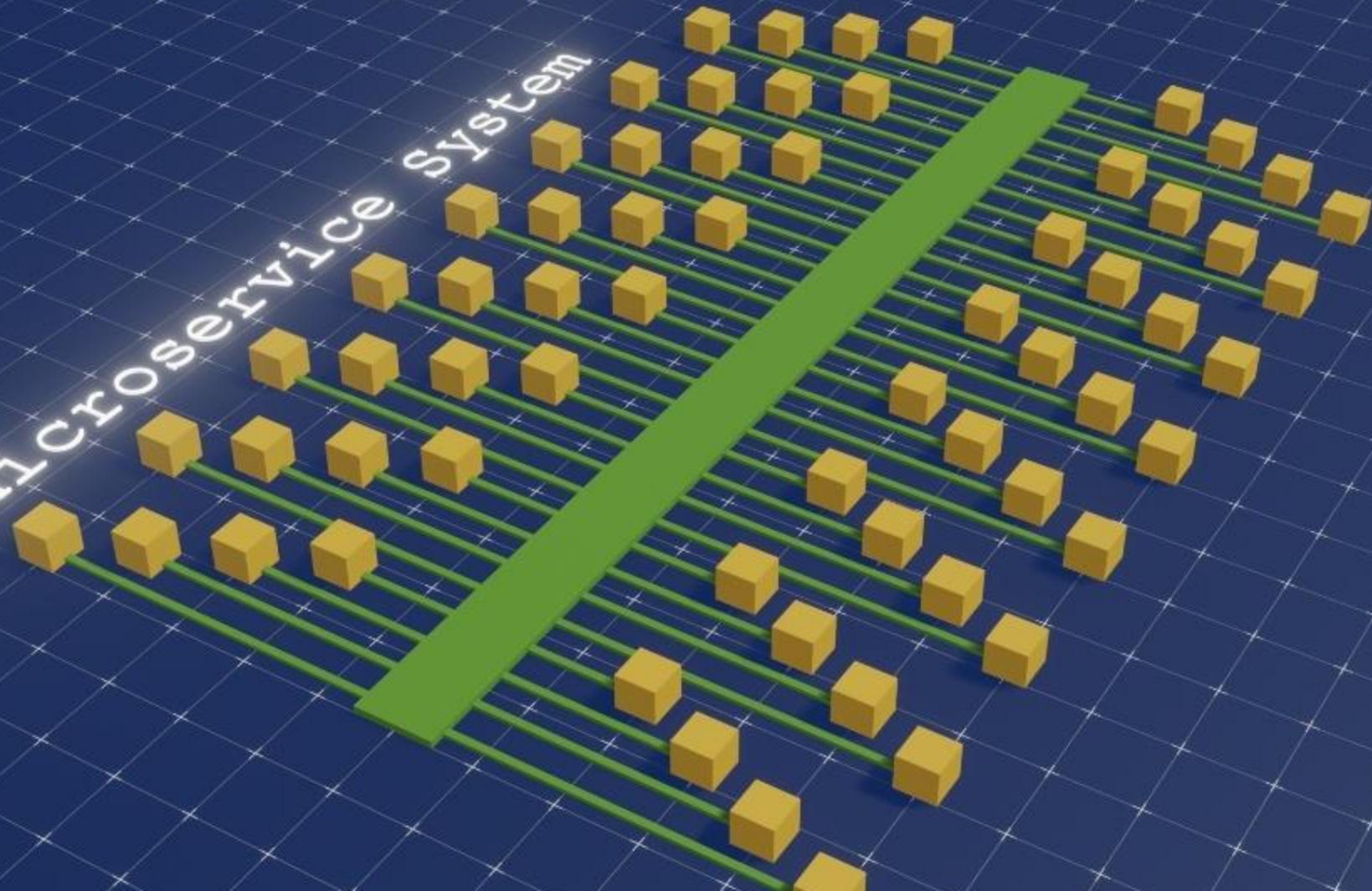
microservice
system

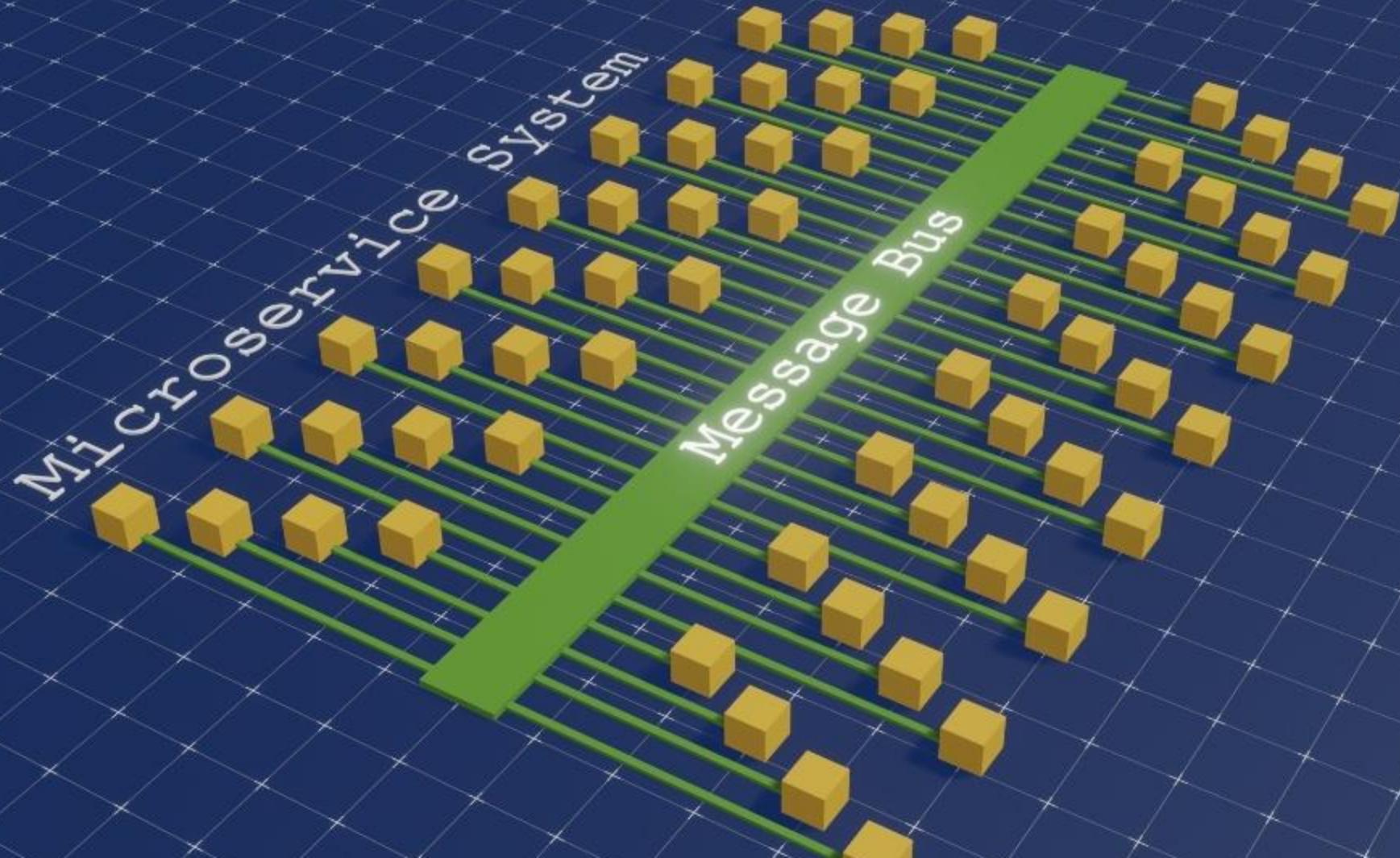


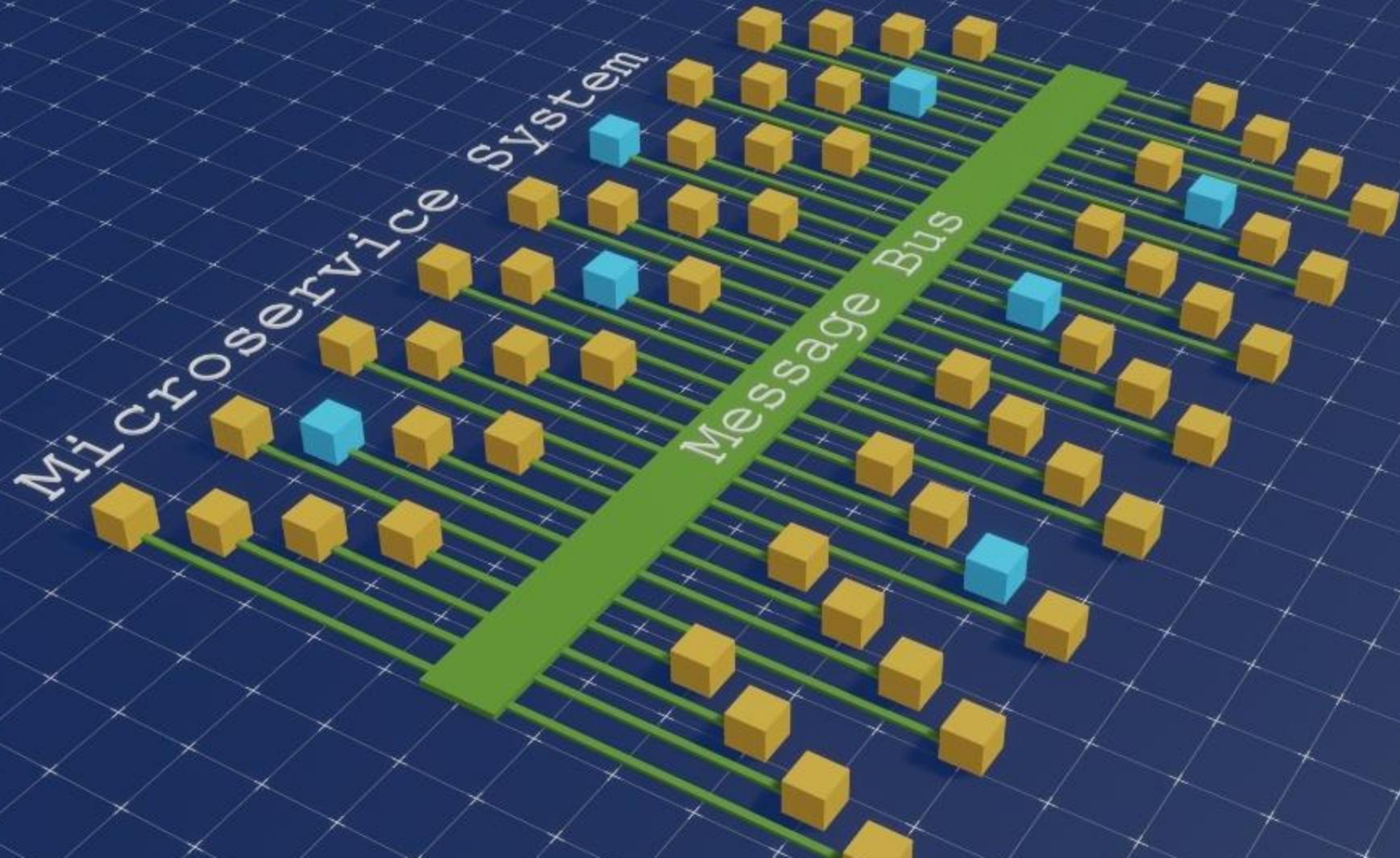


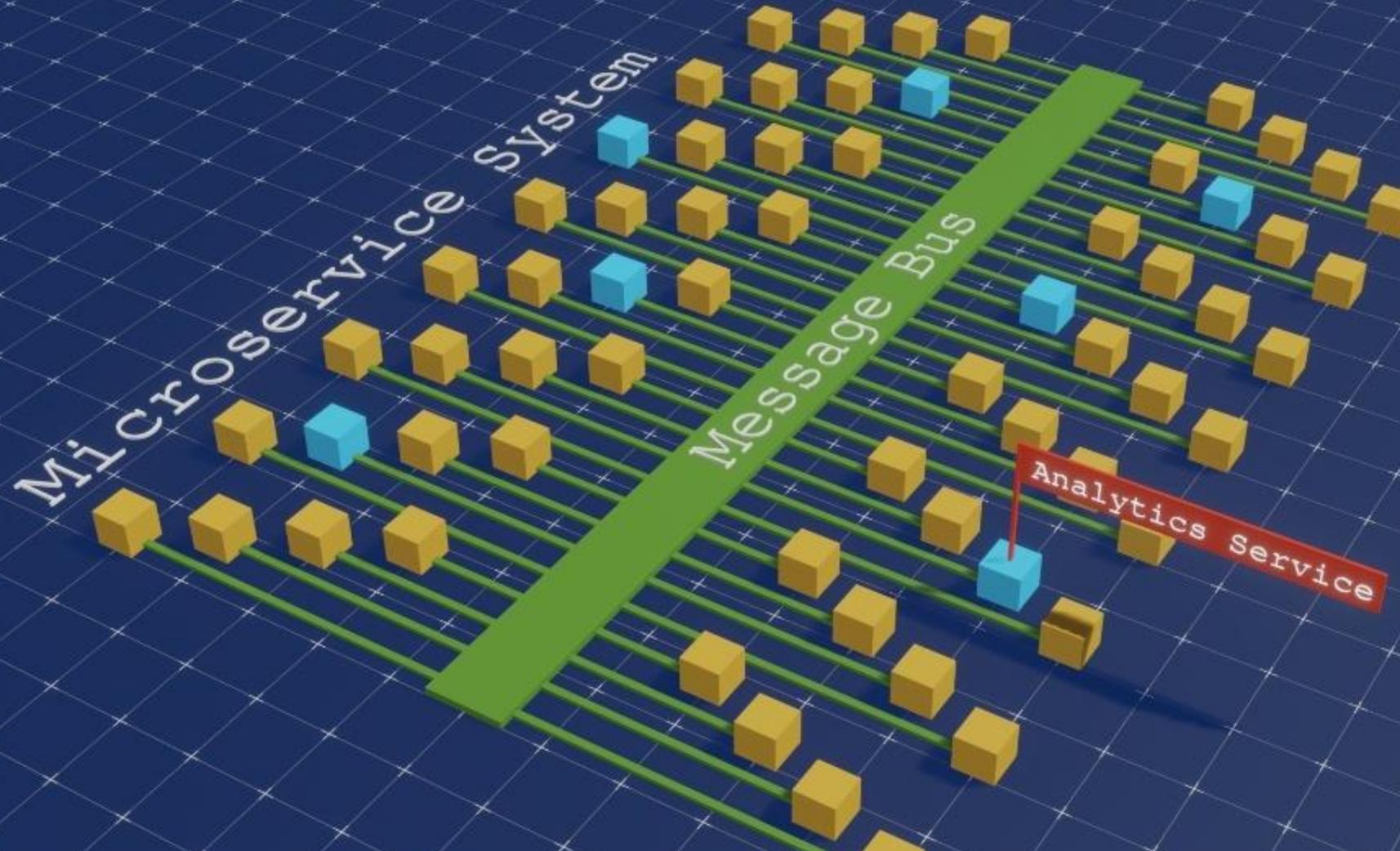


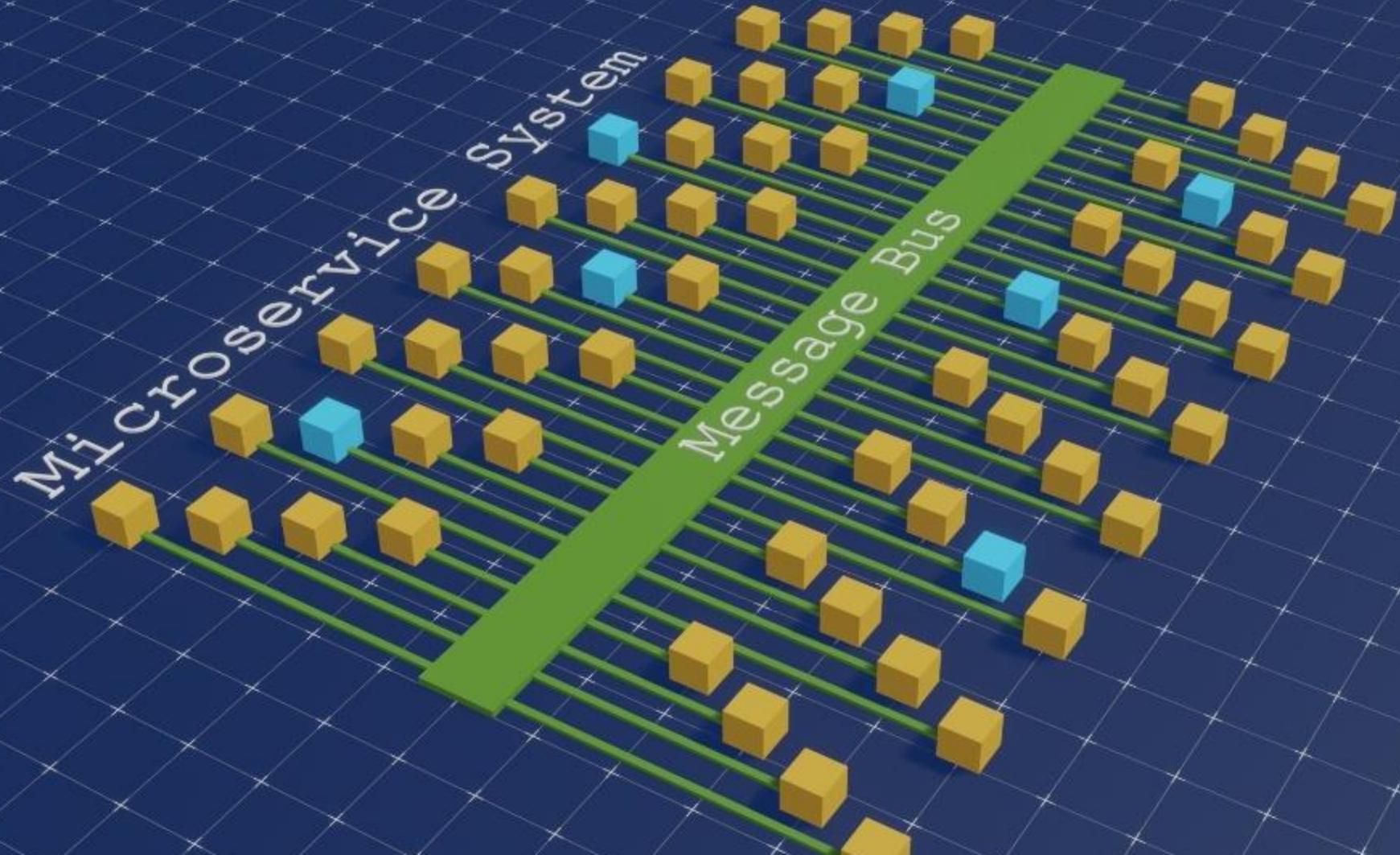
Microservice System

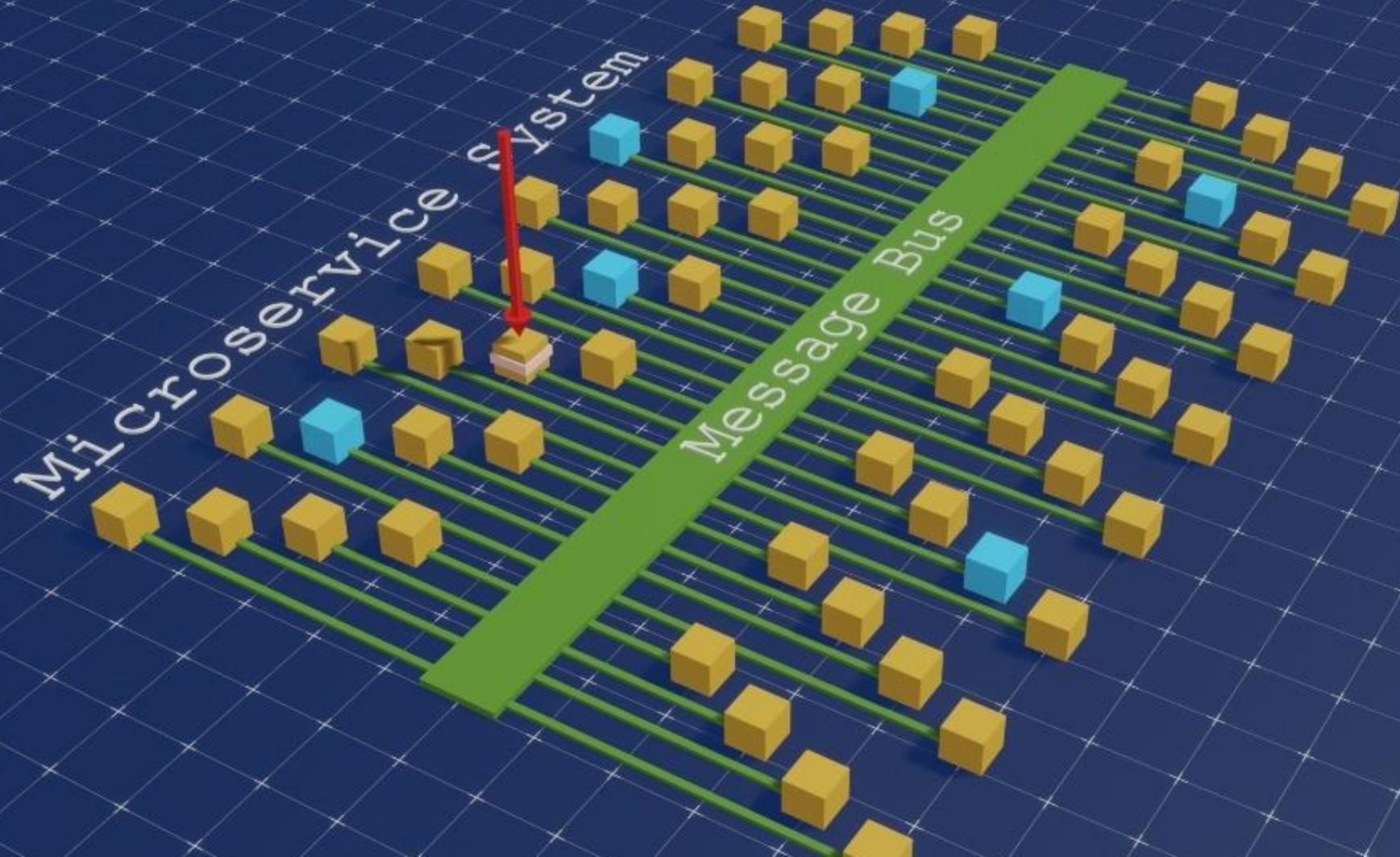


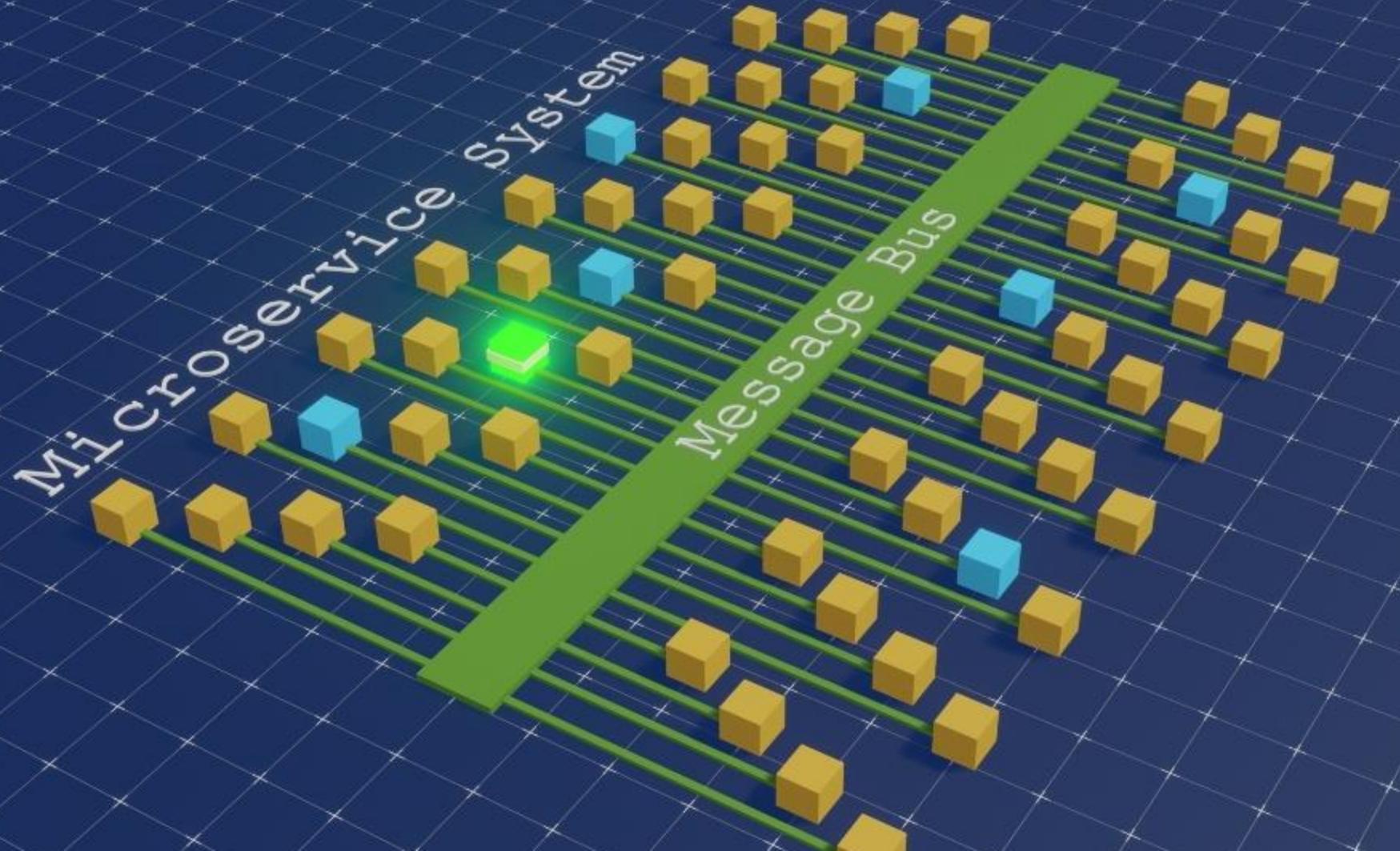


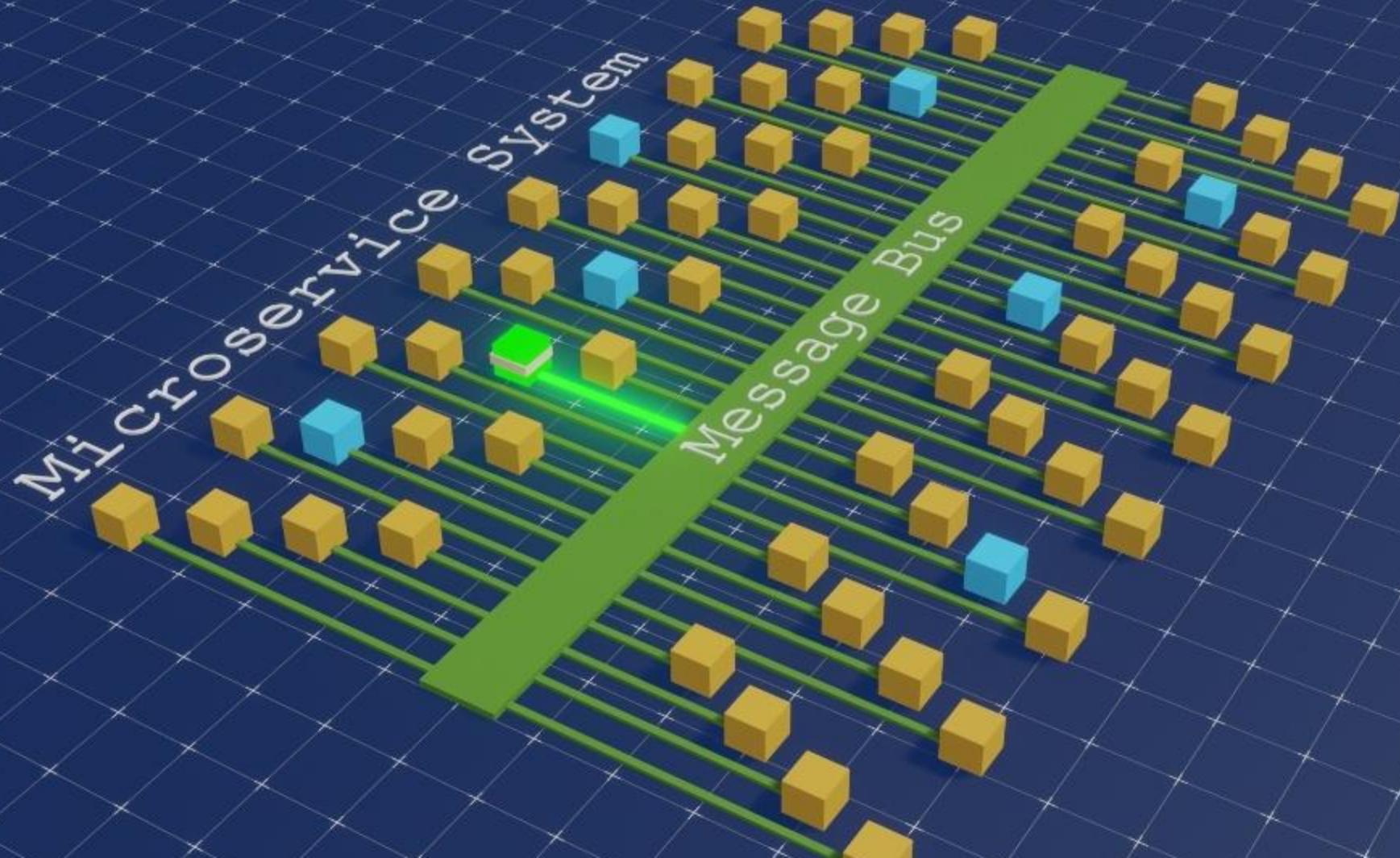


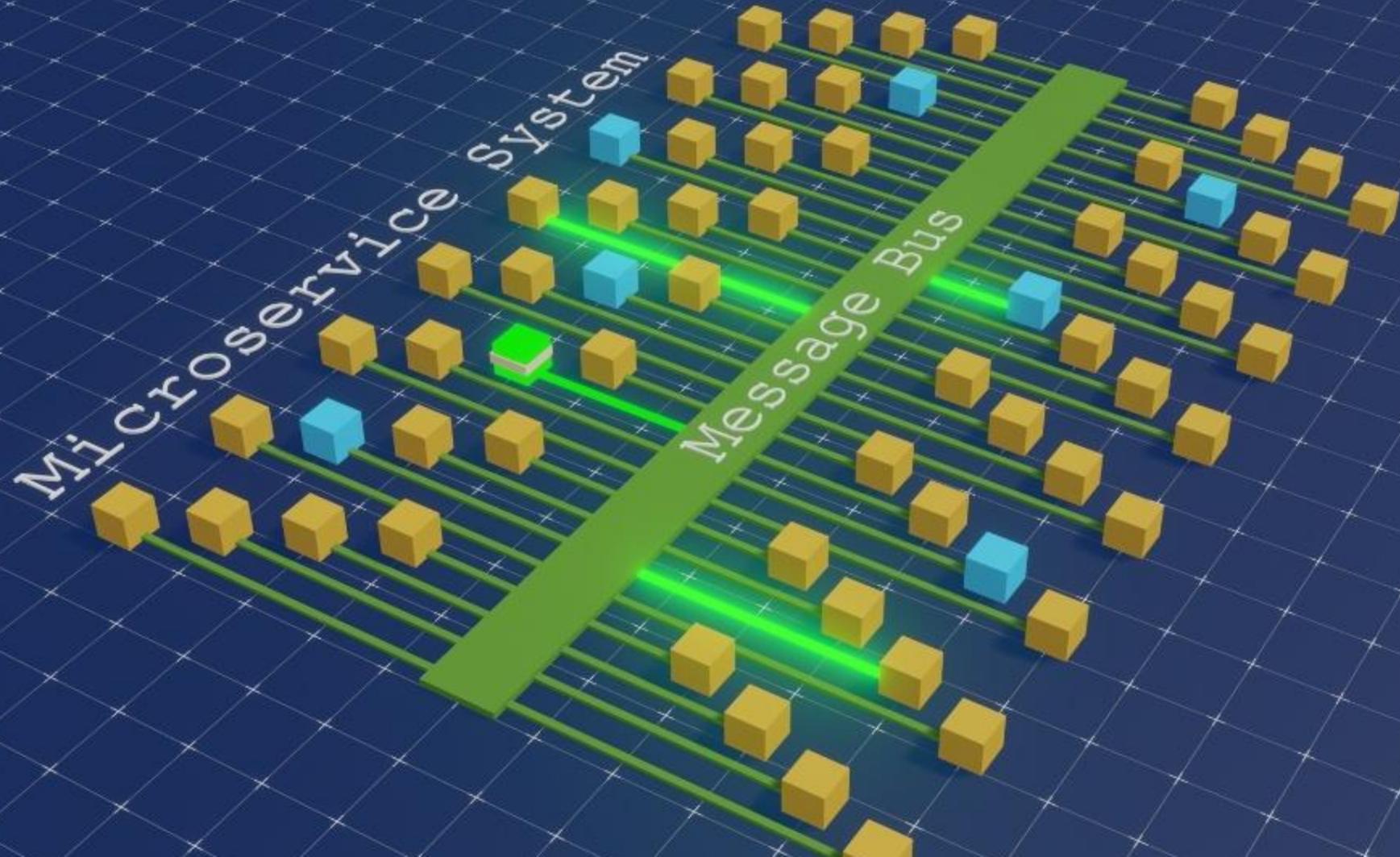


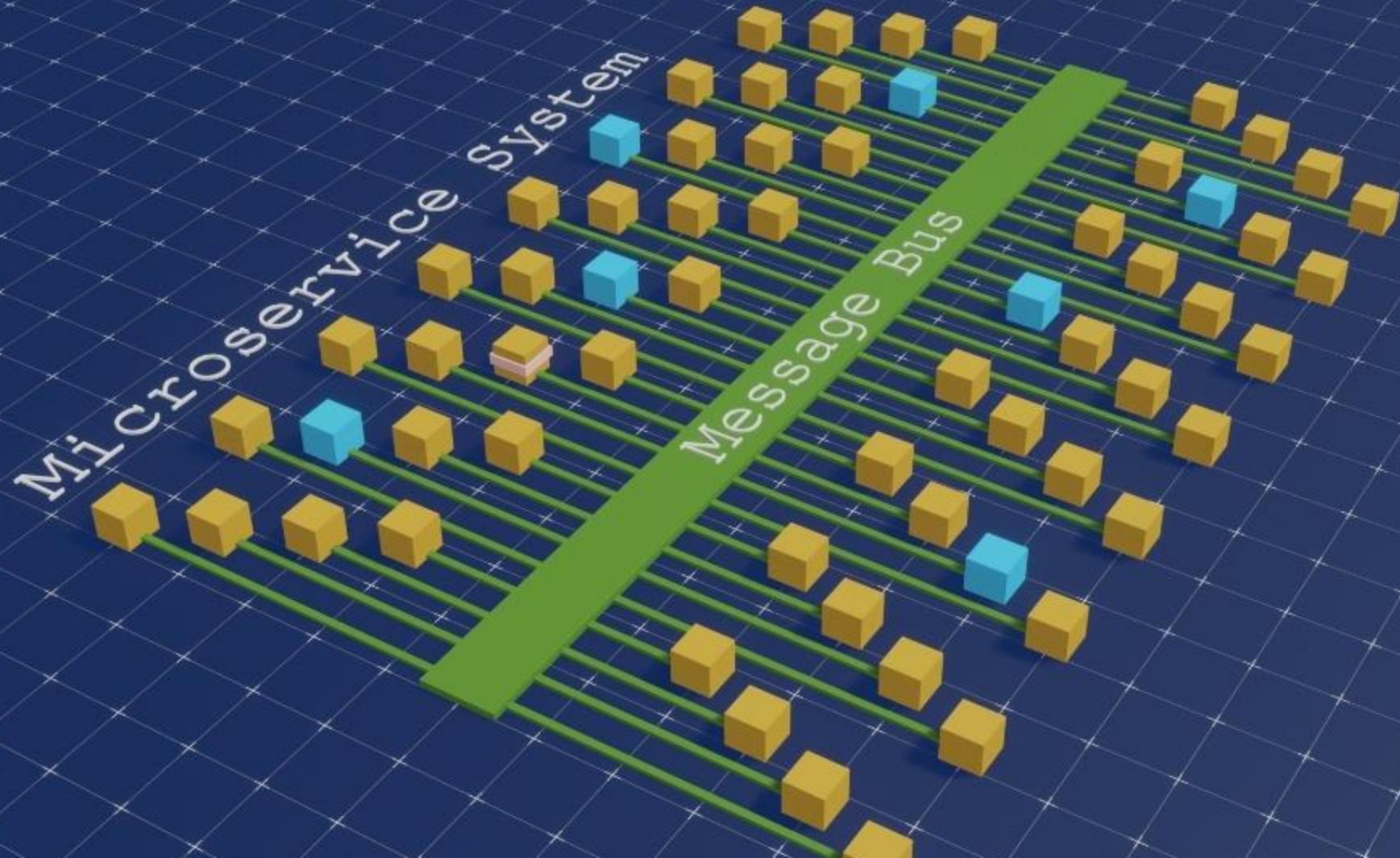


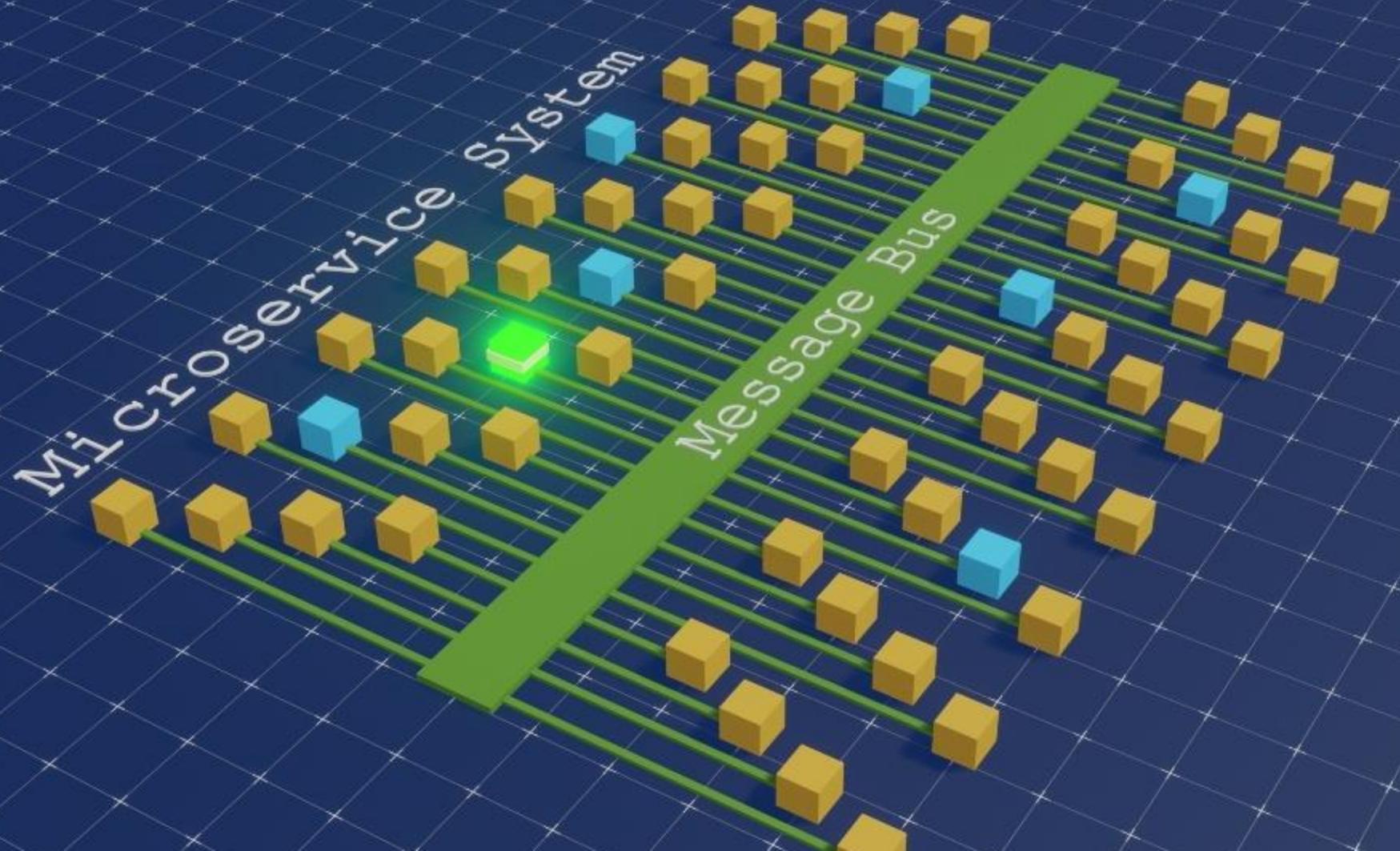


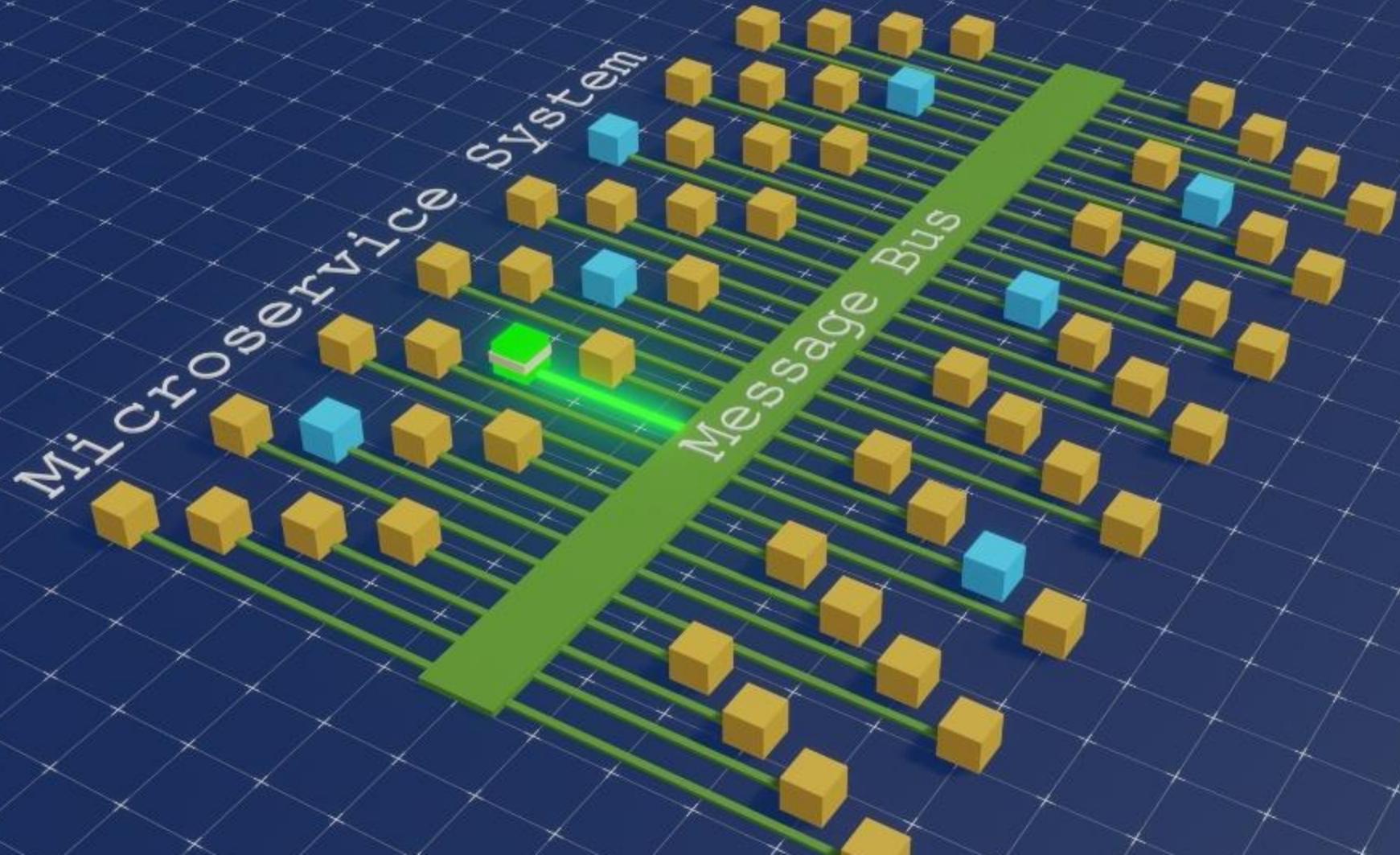


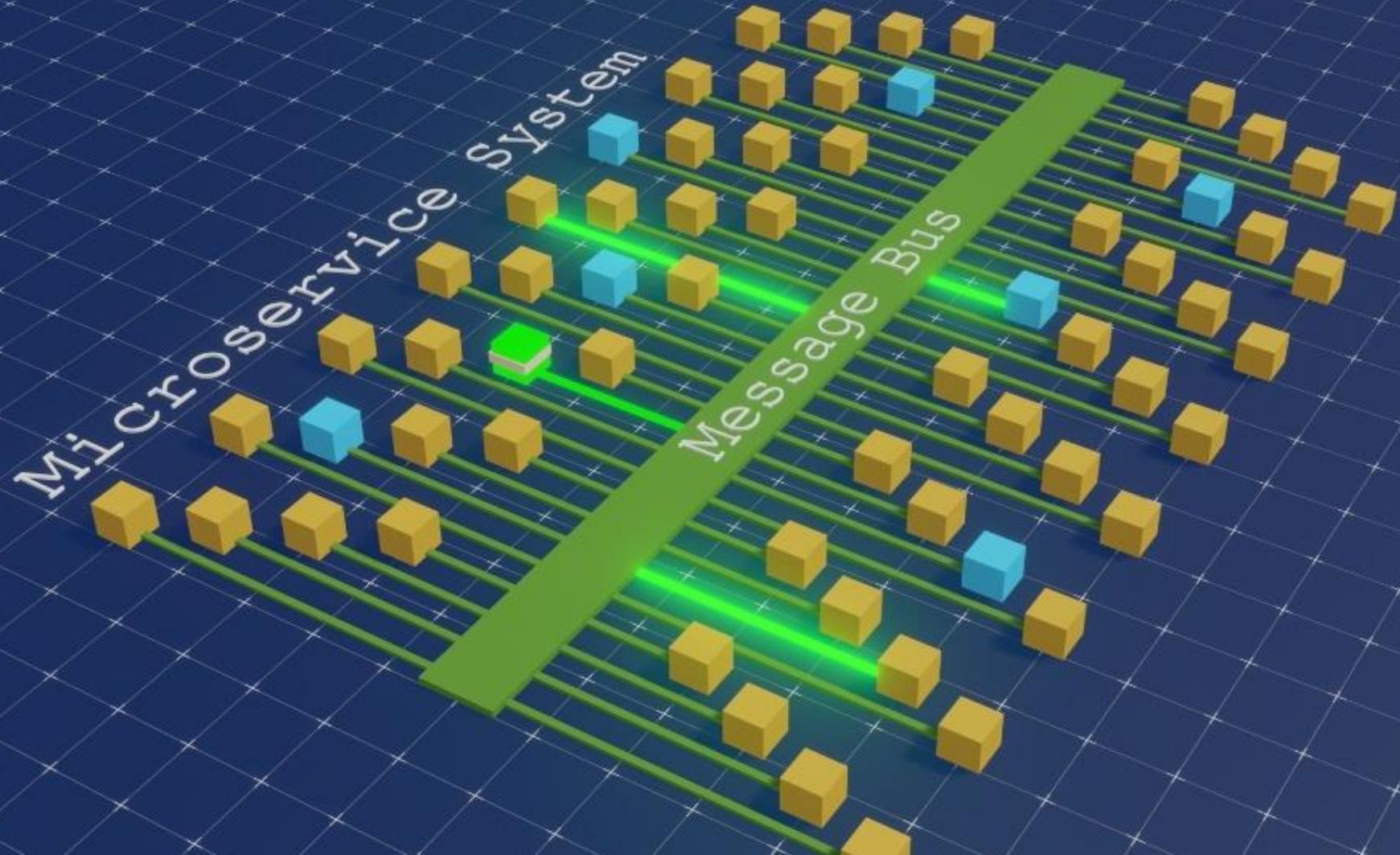


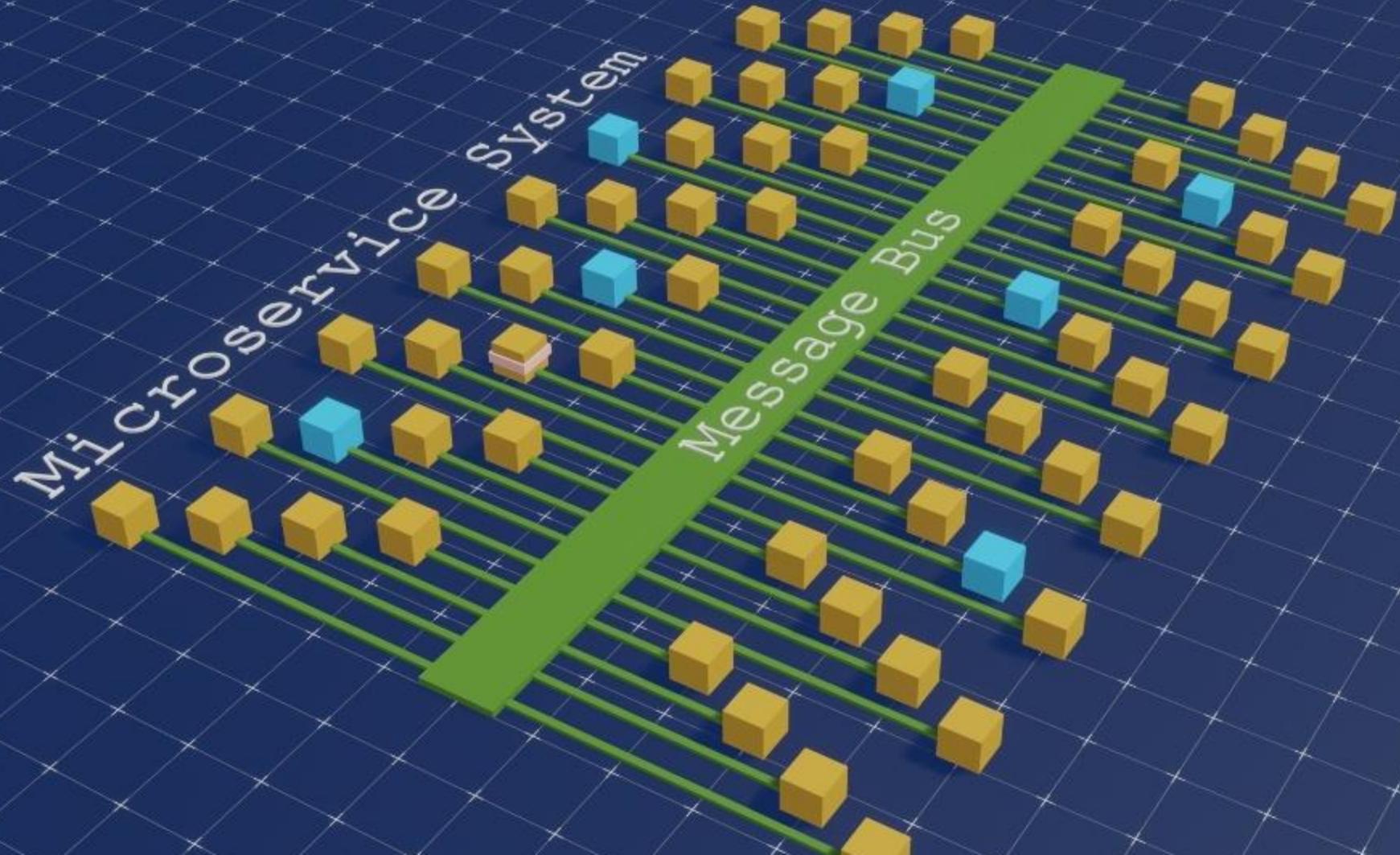


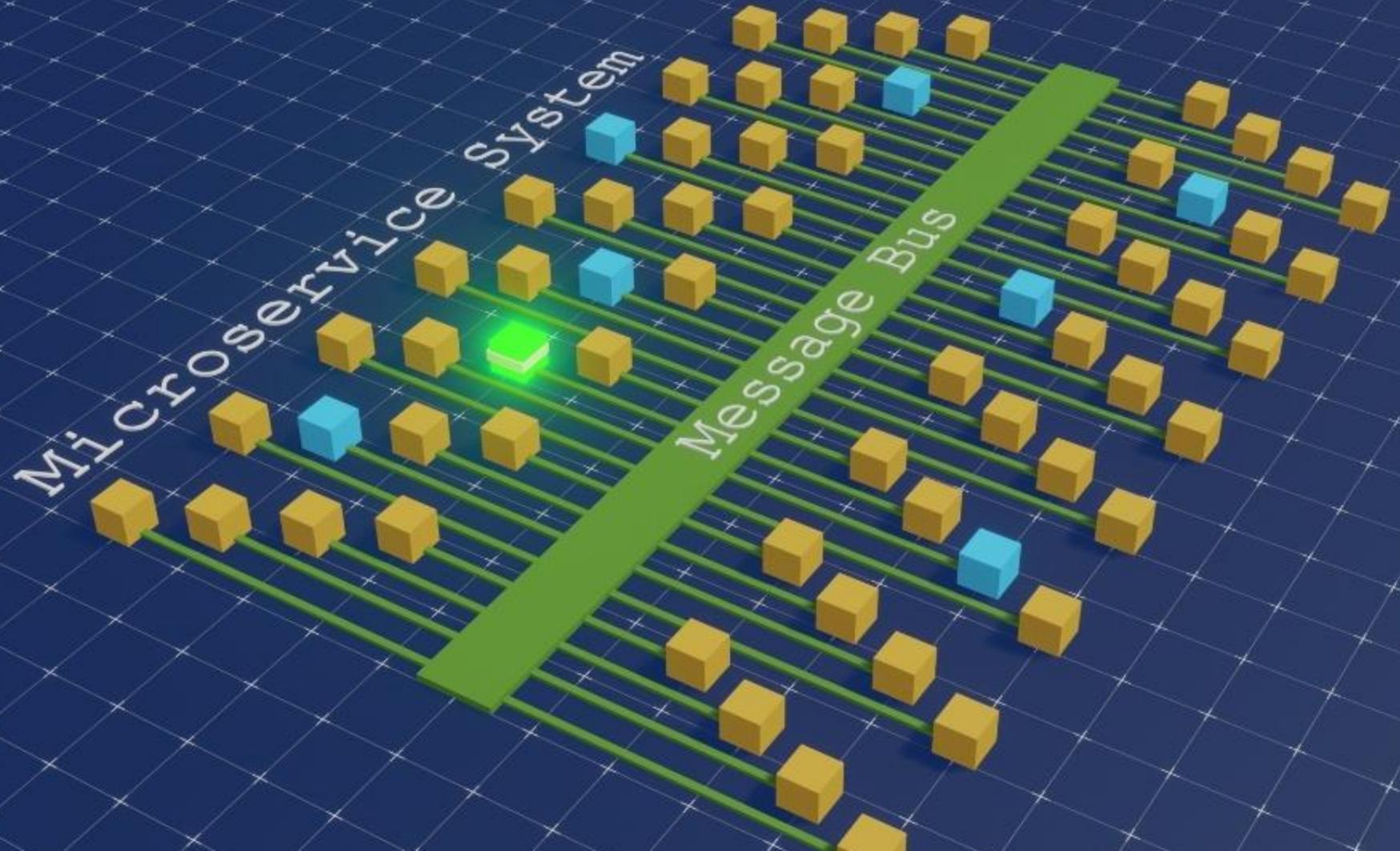


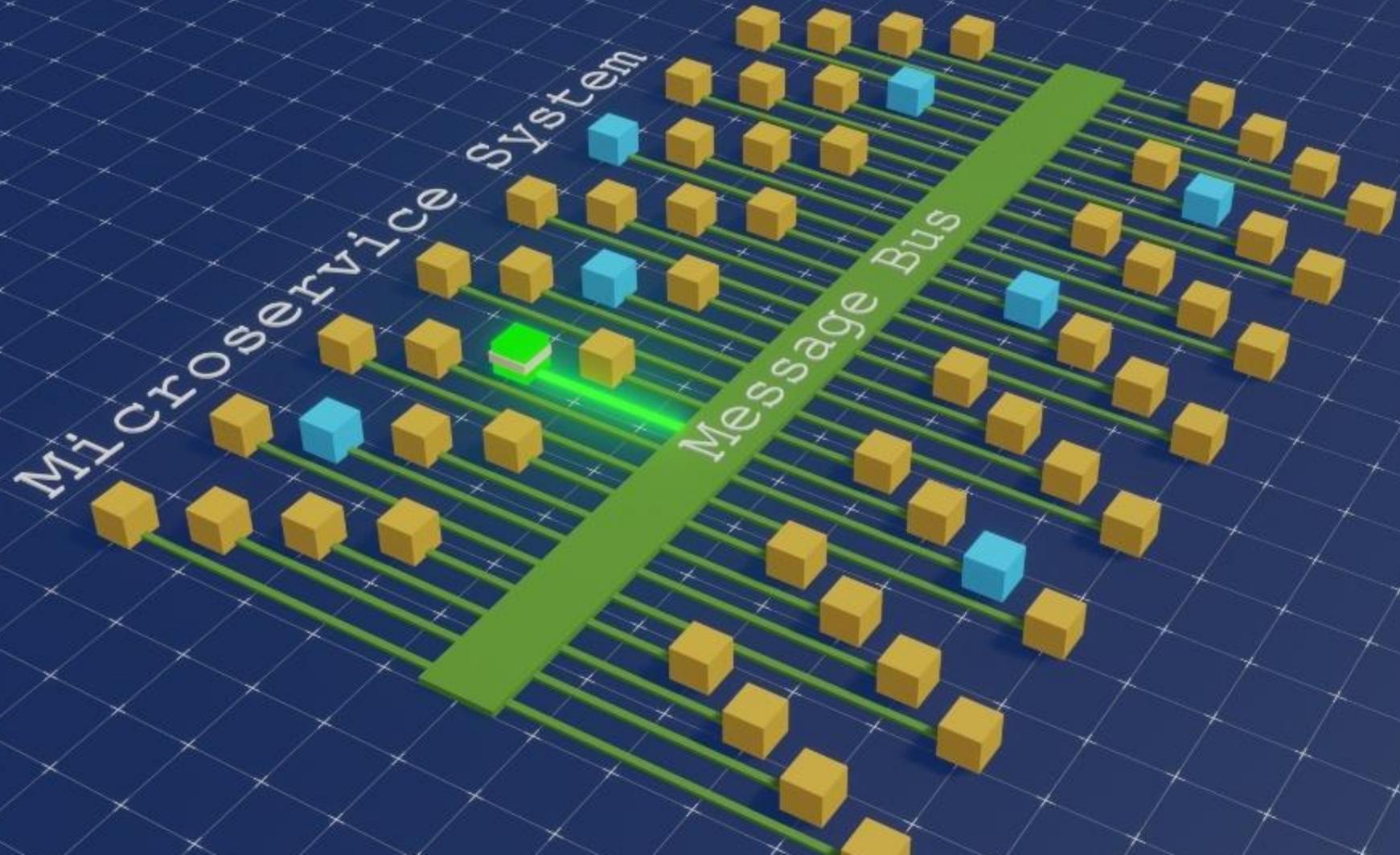


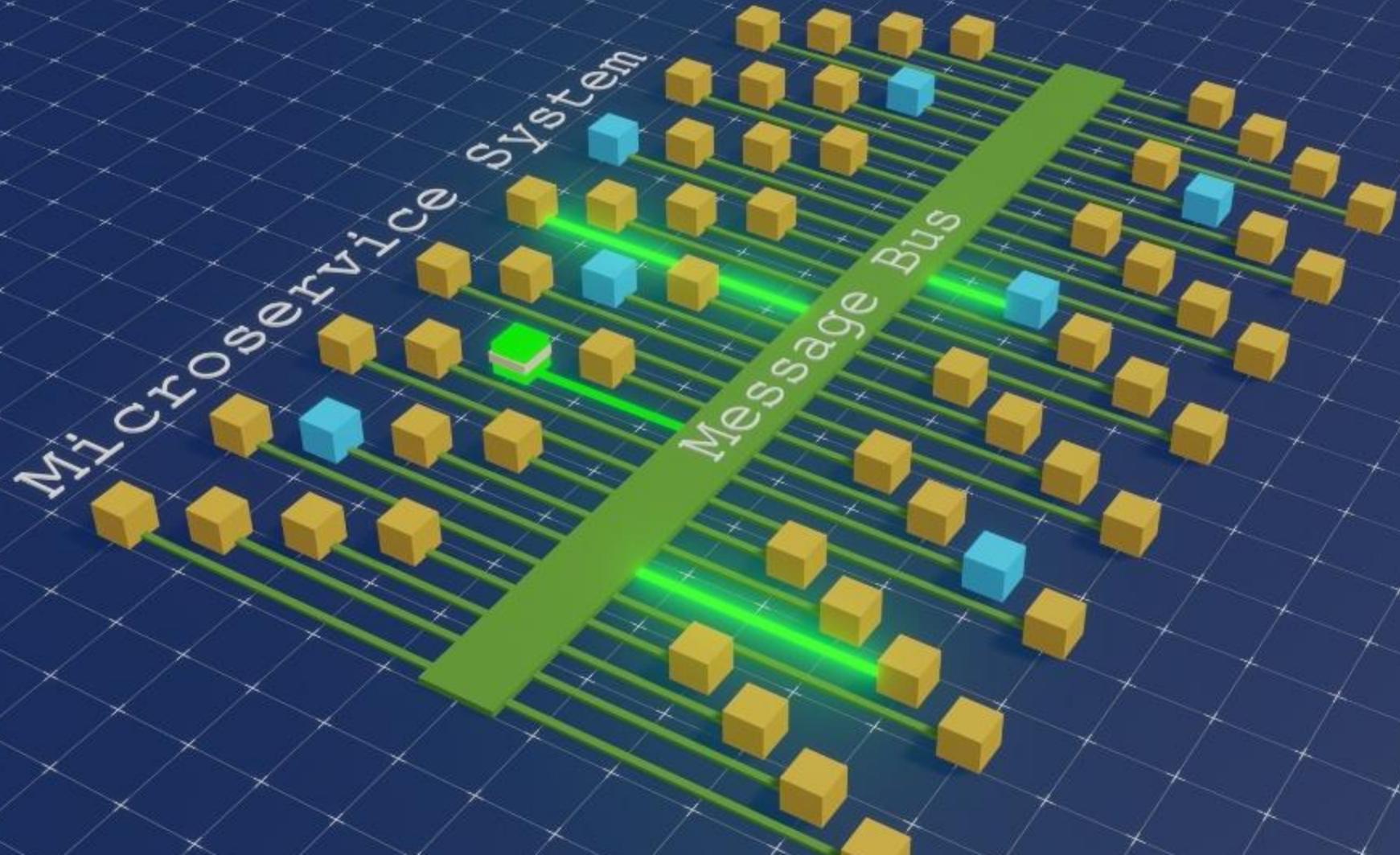


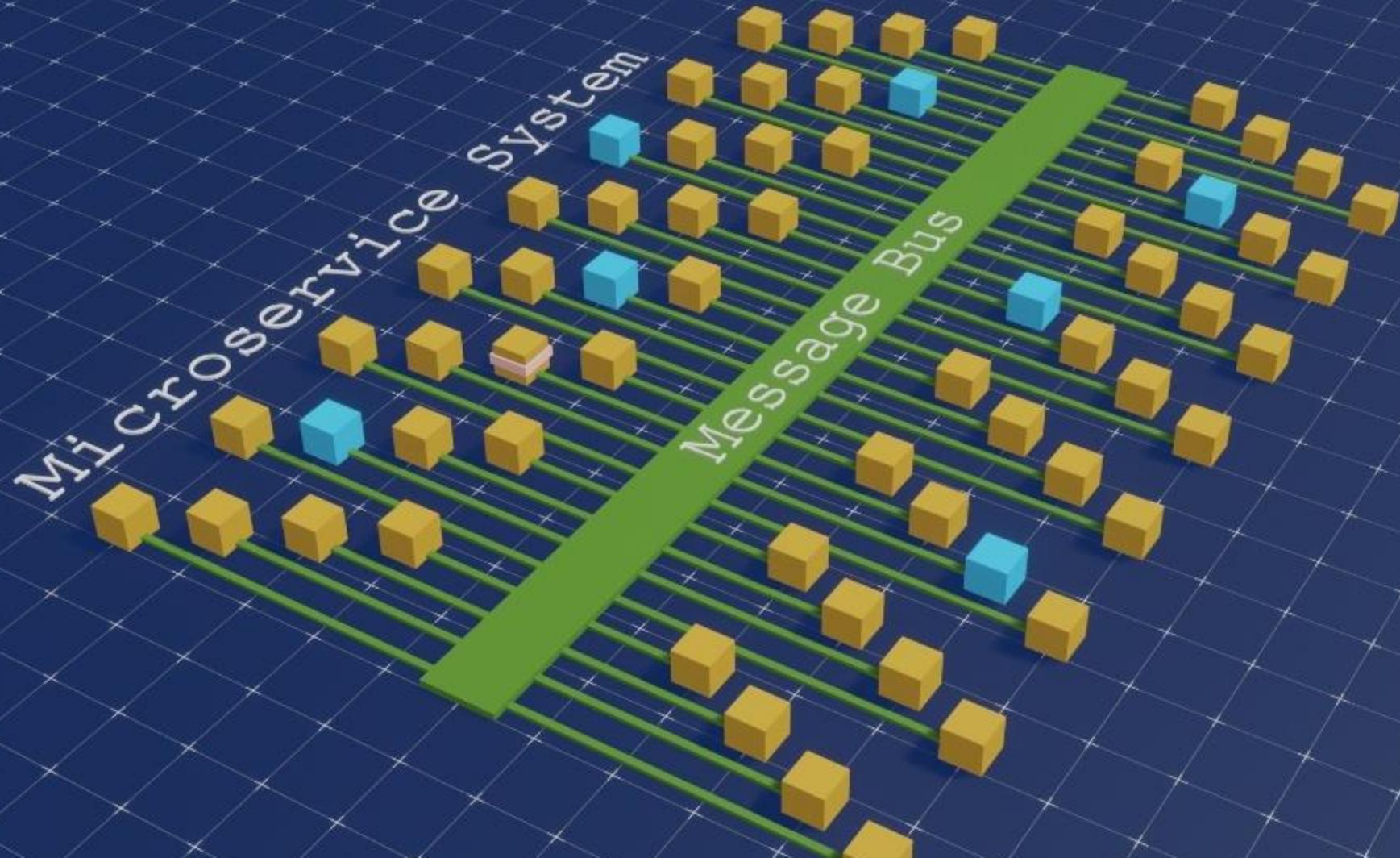


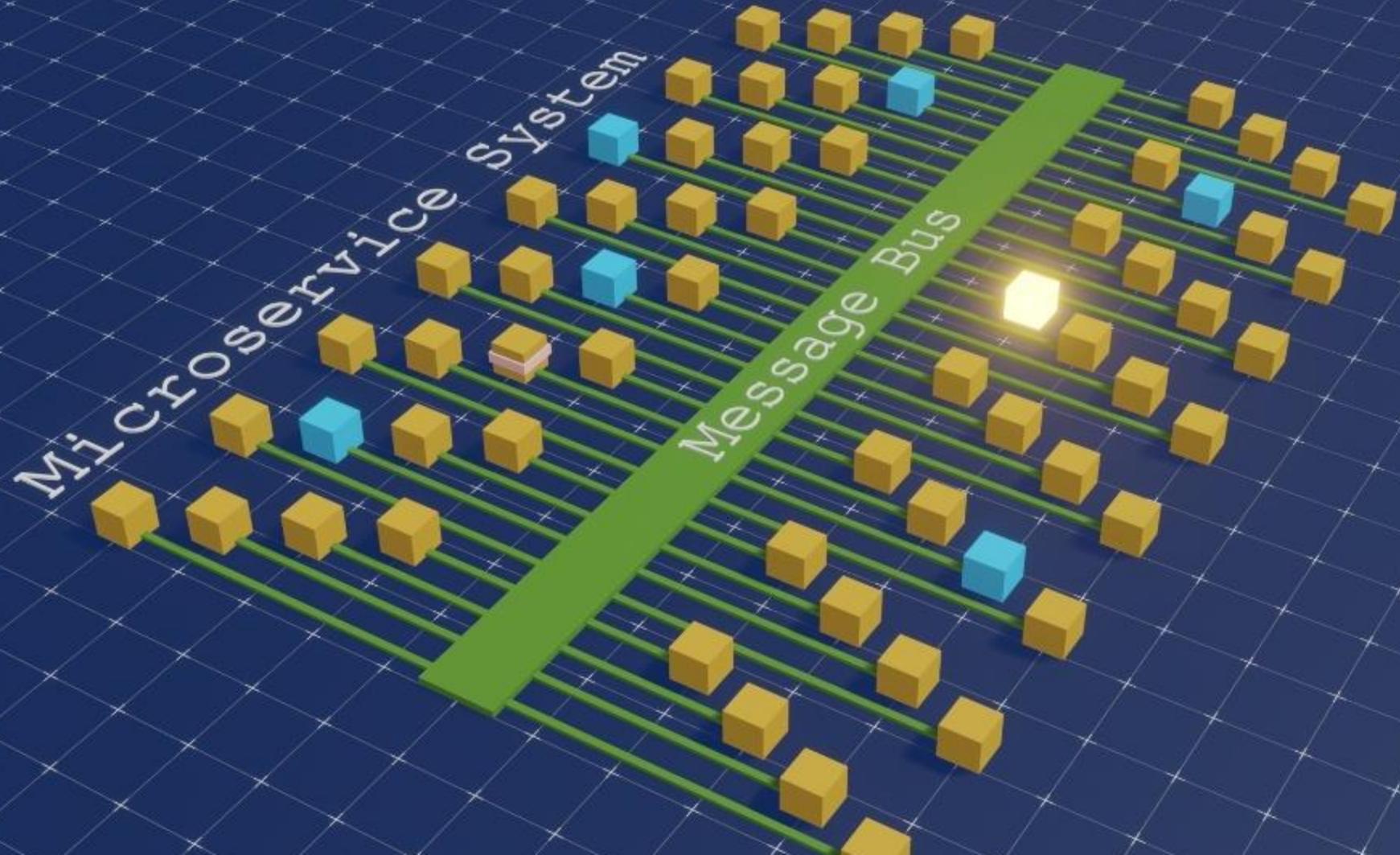


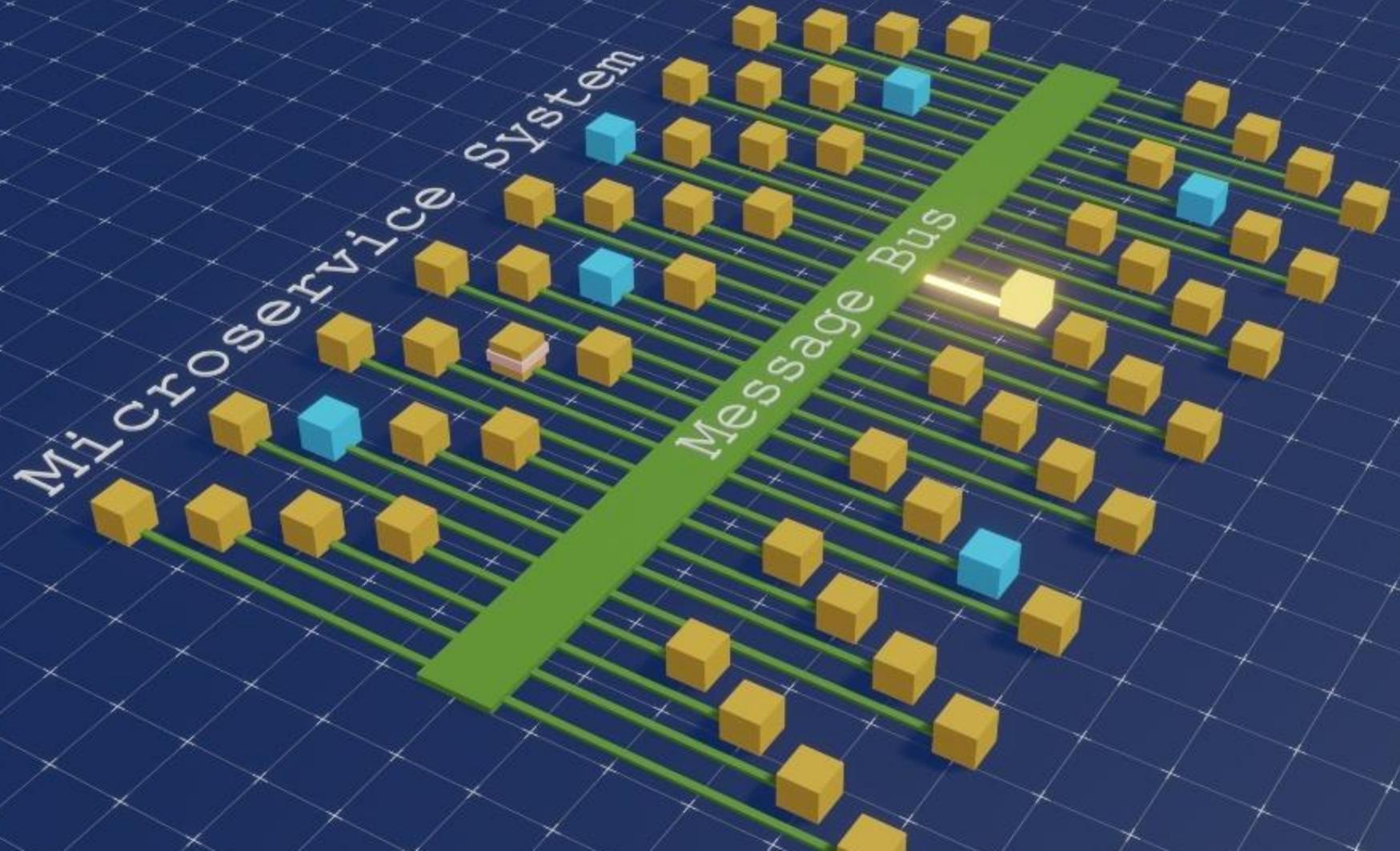


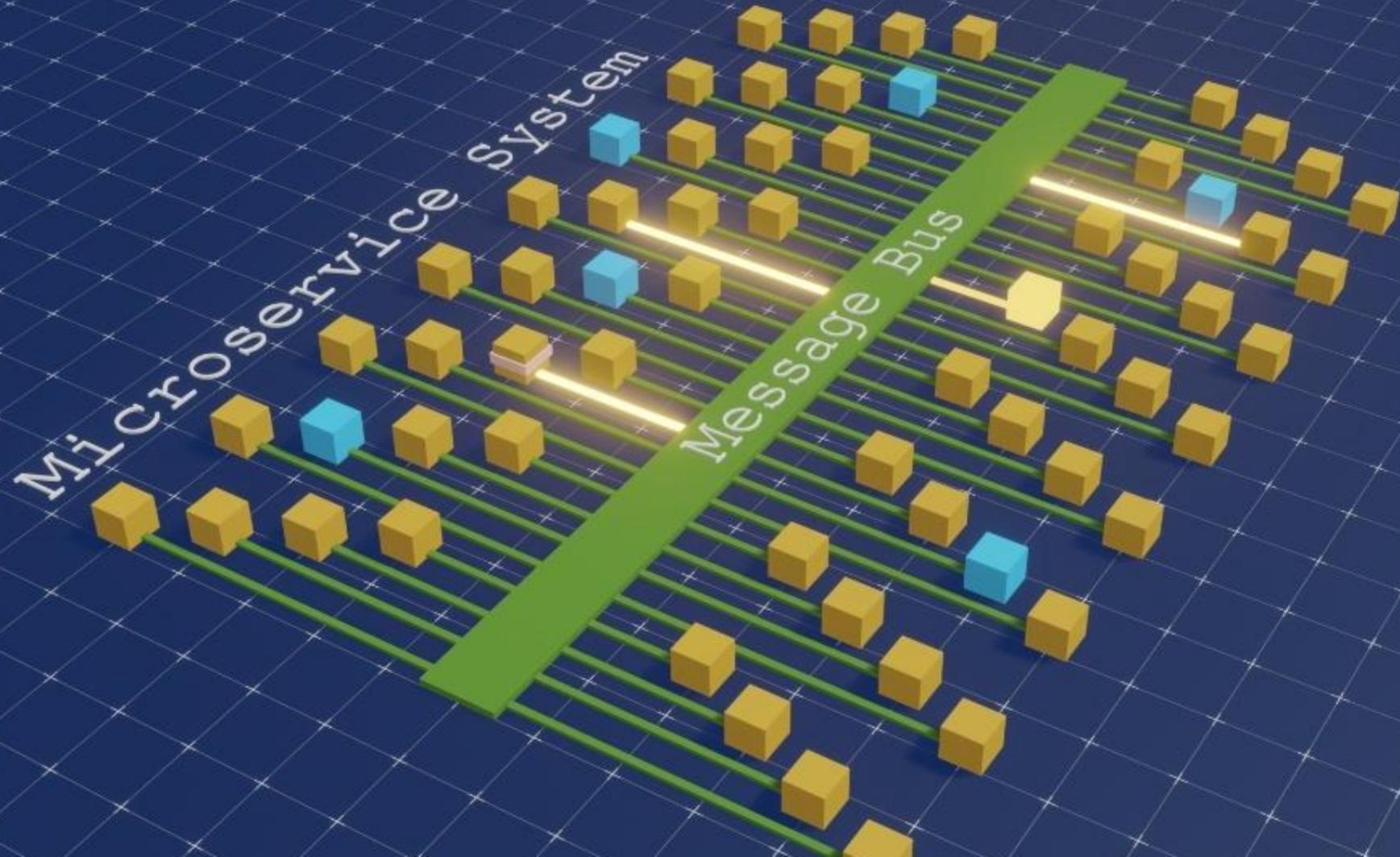


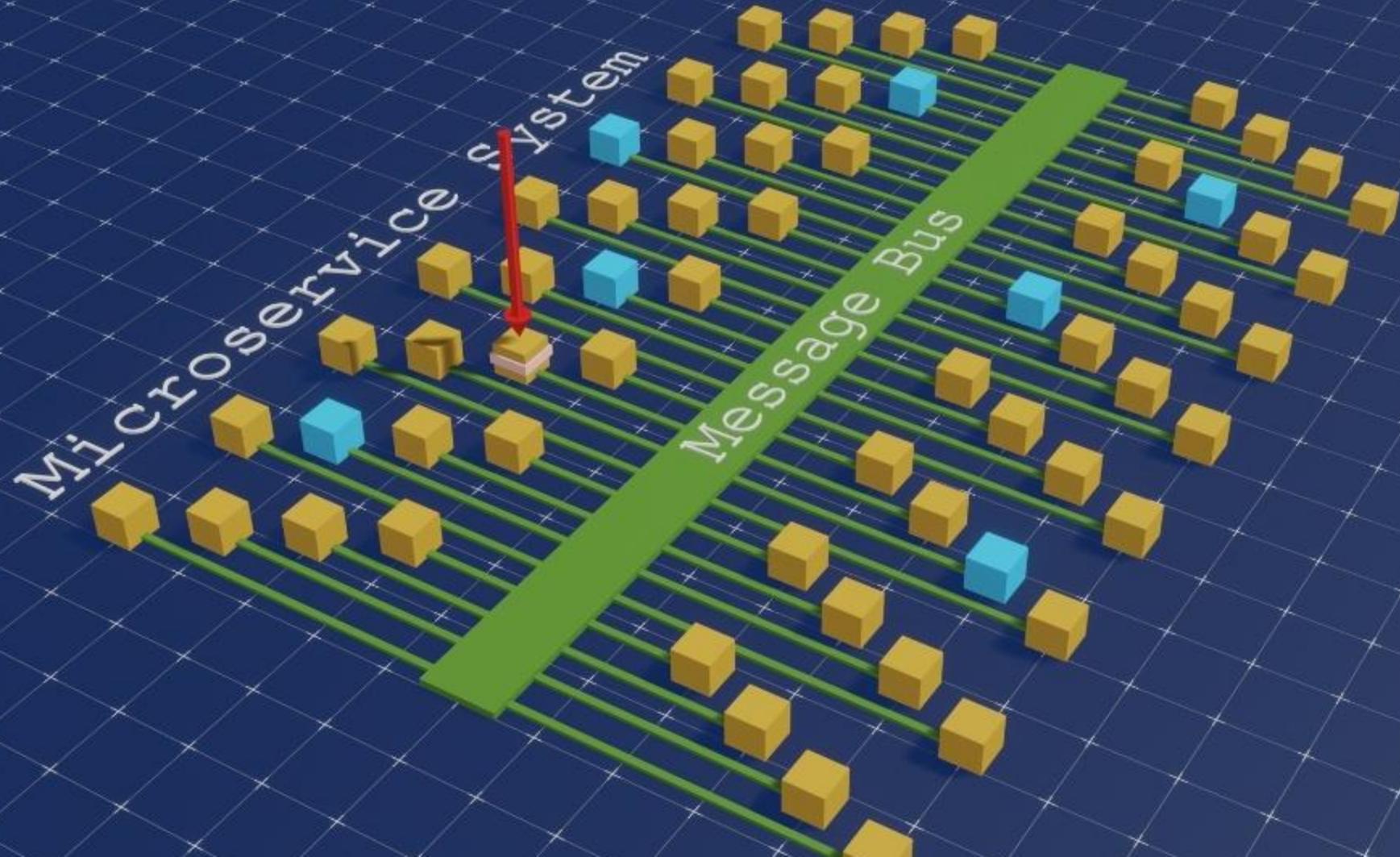


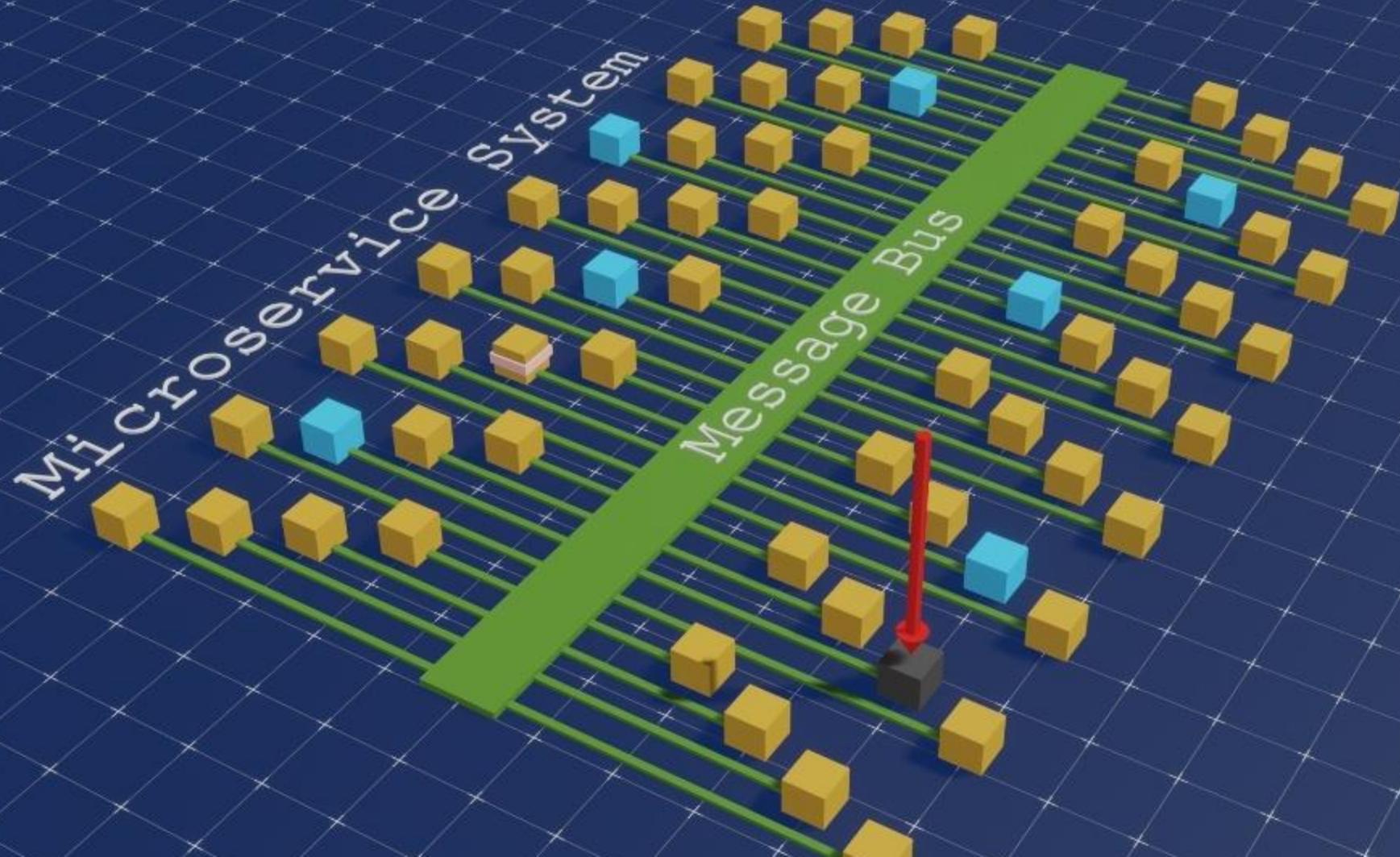


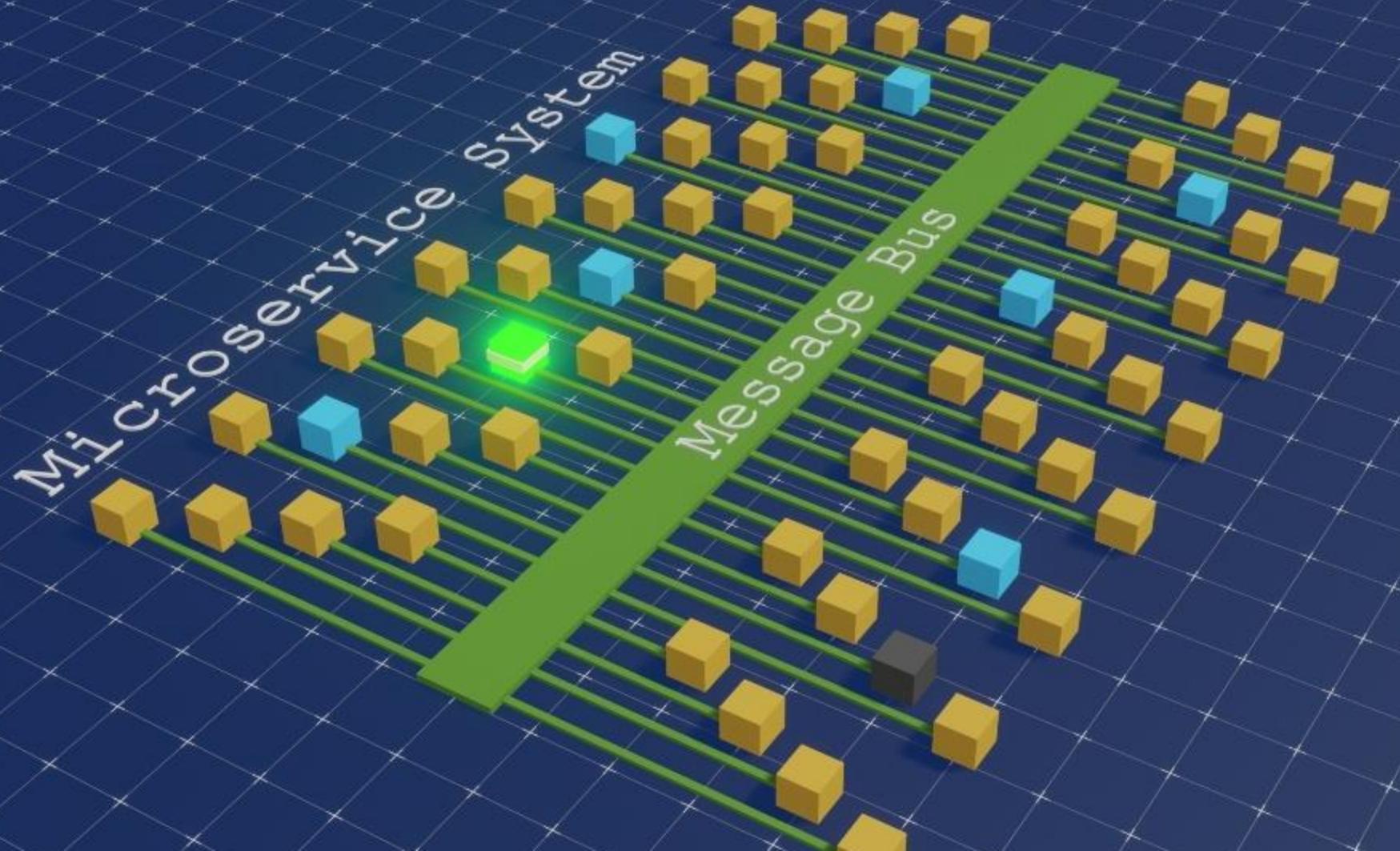


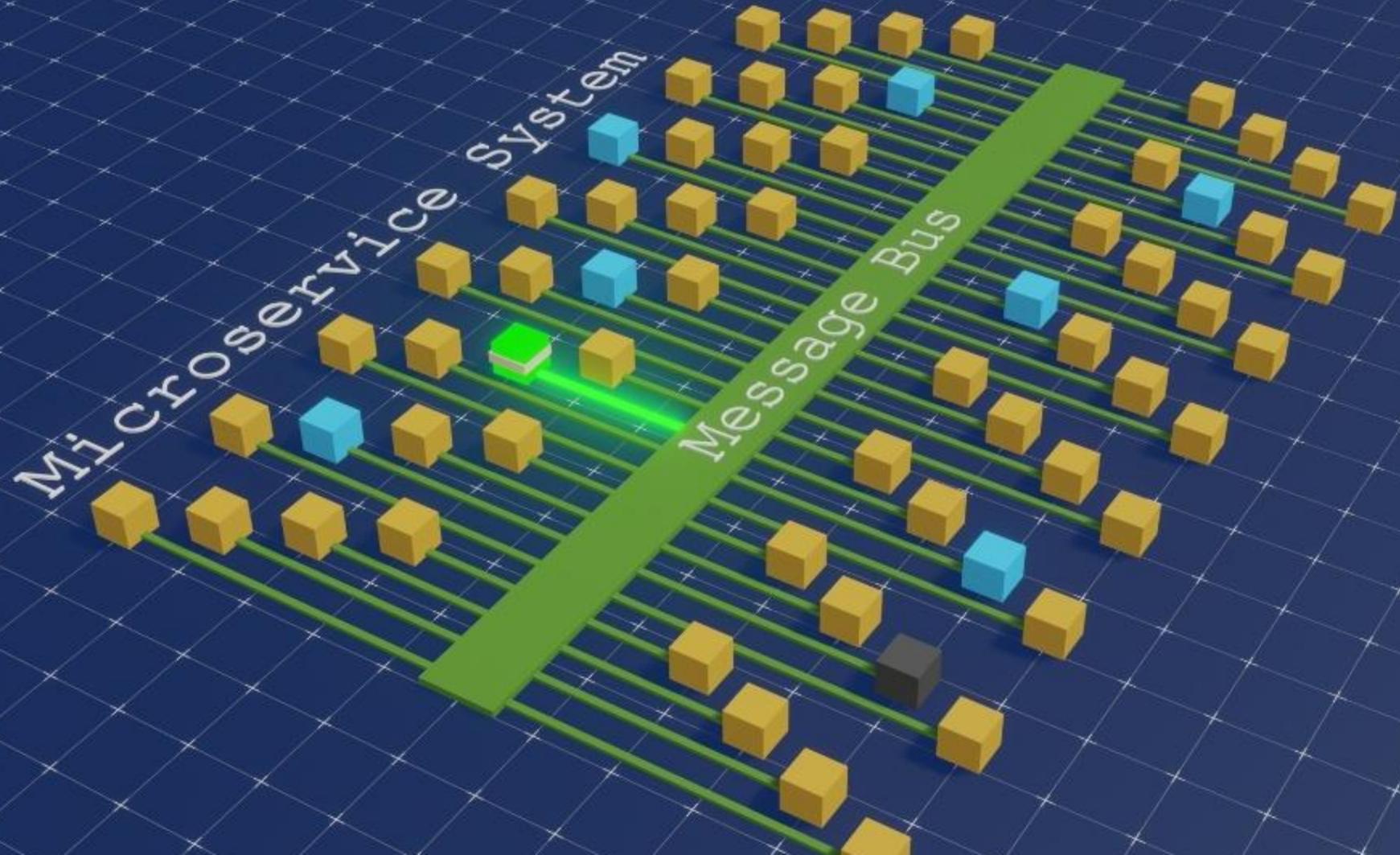


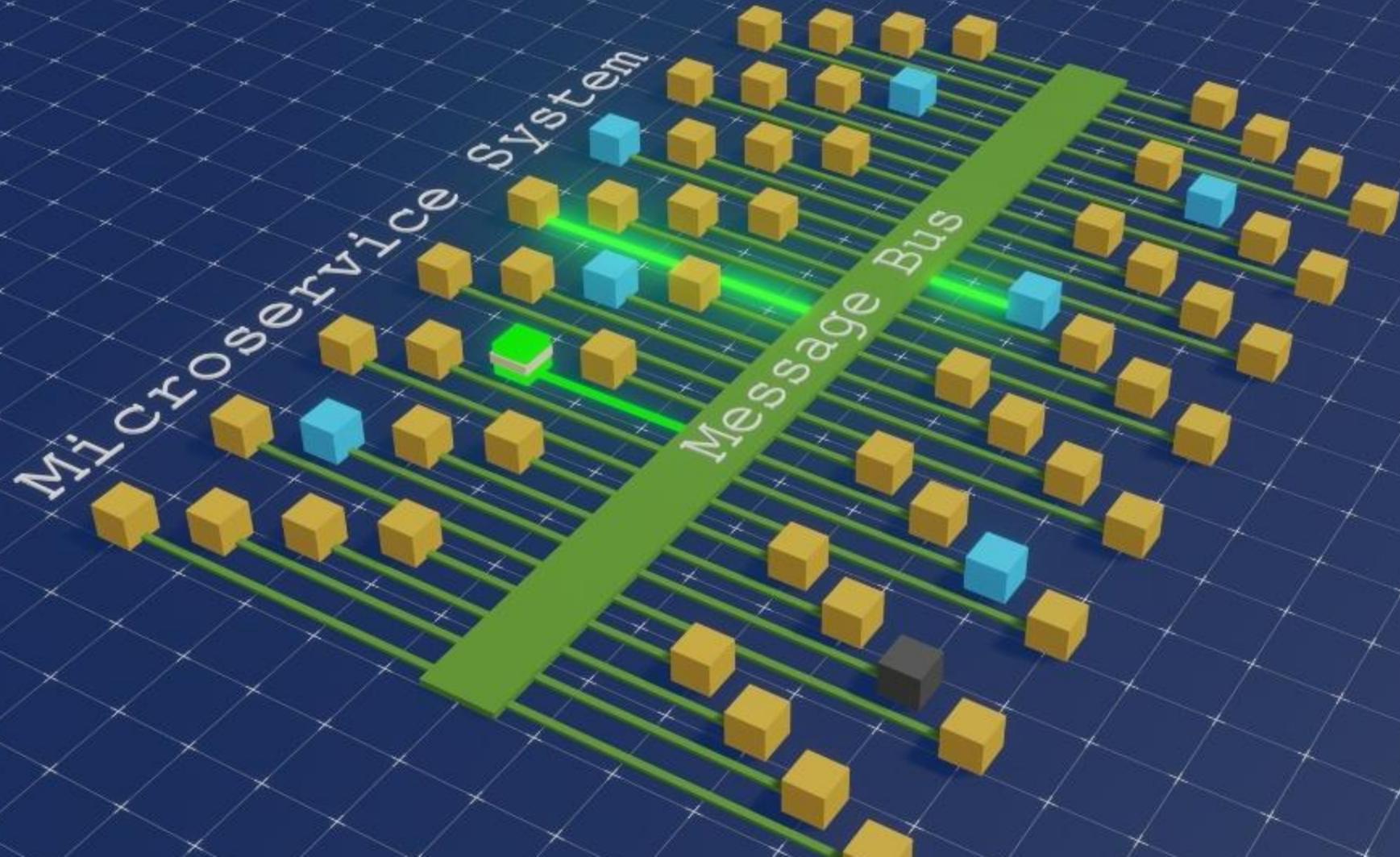


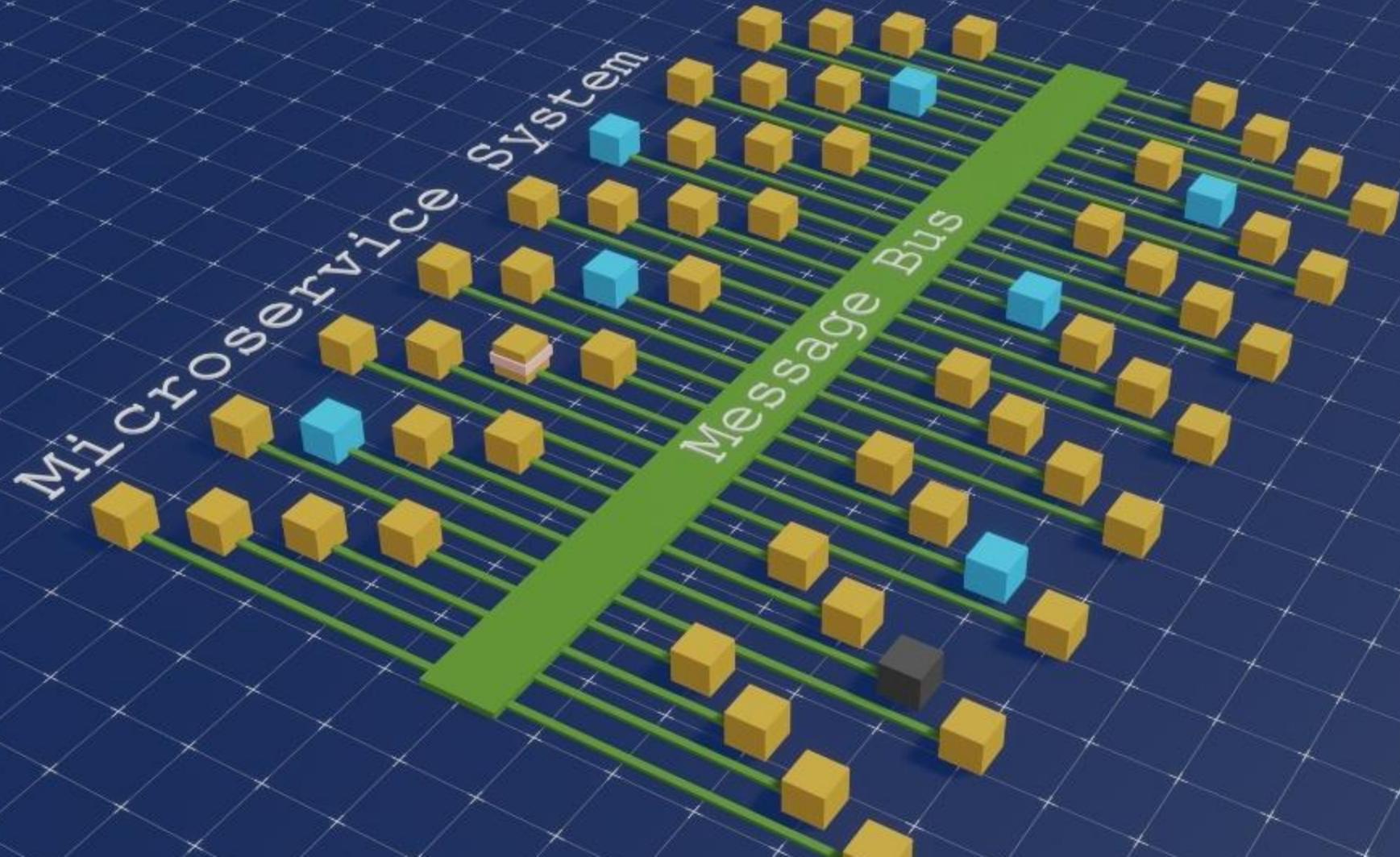


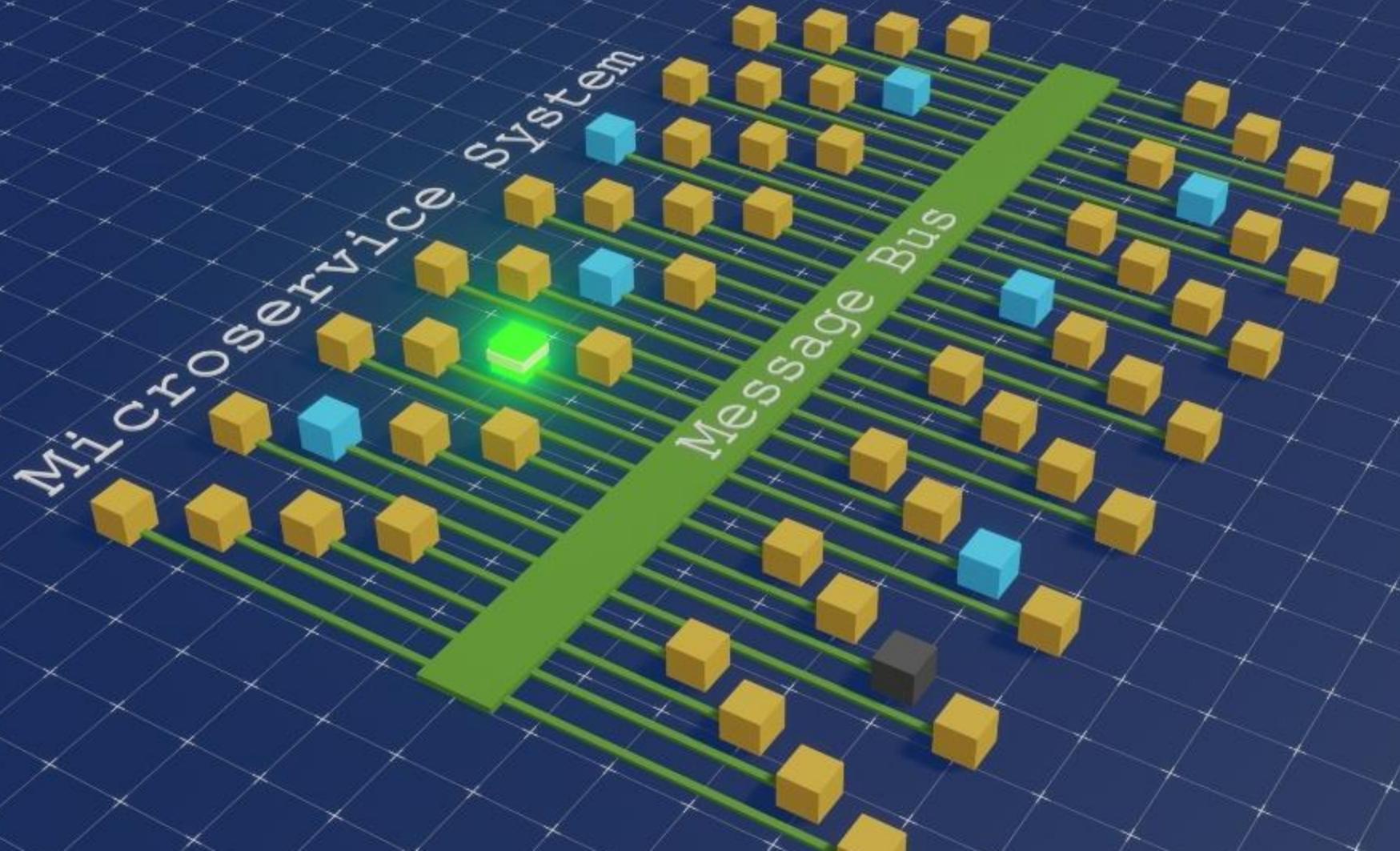


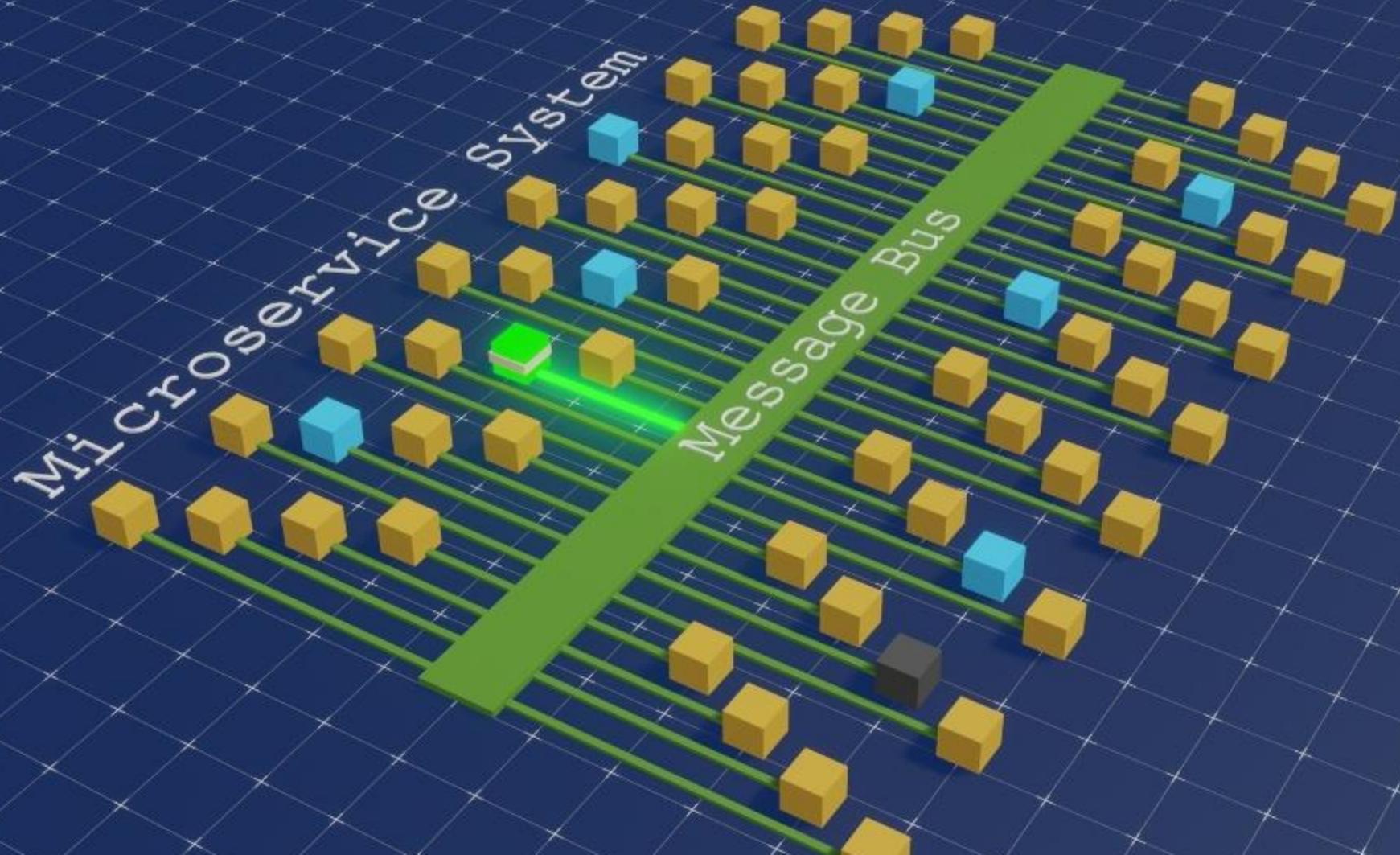


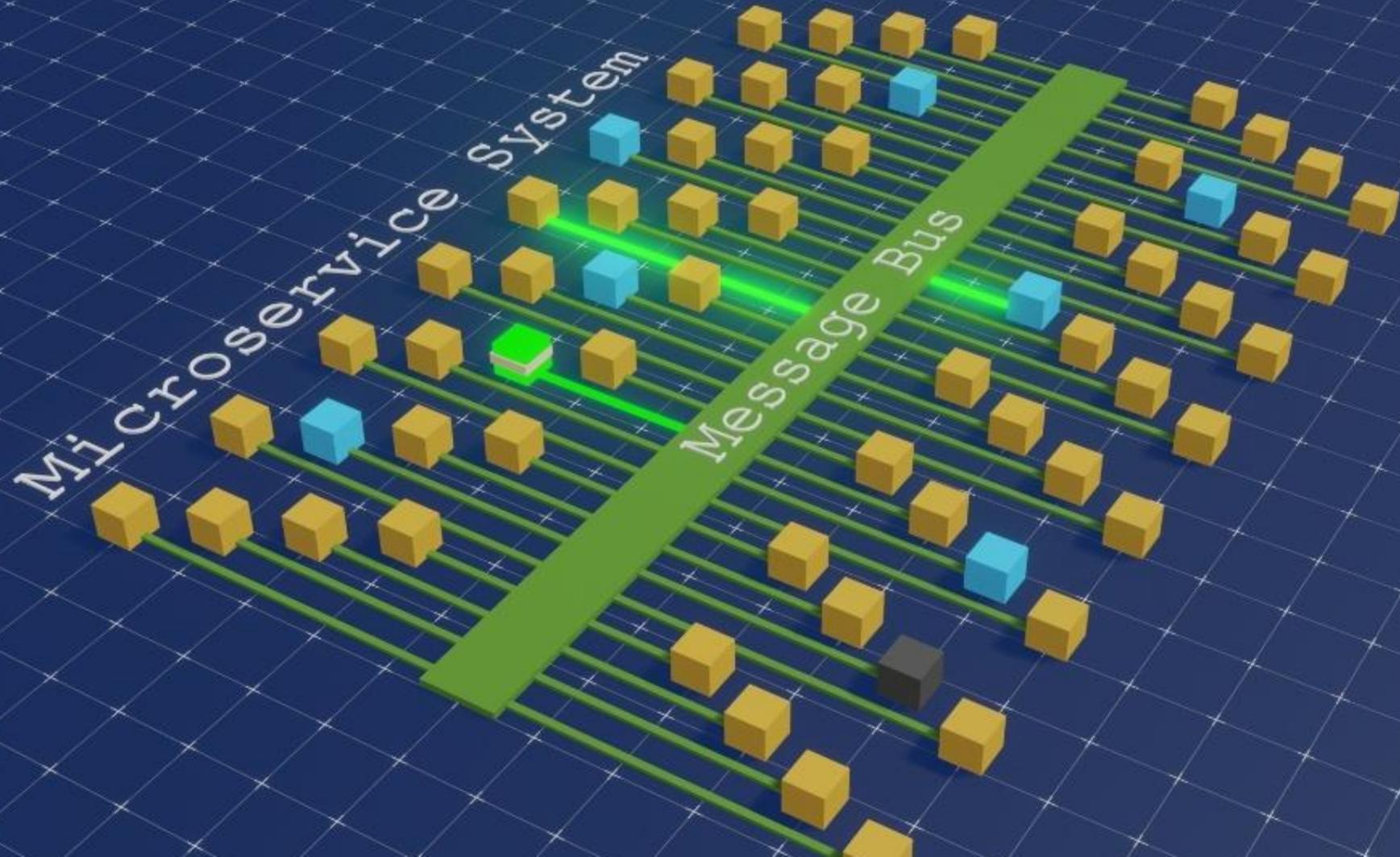


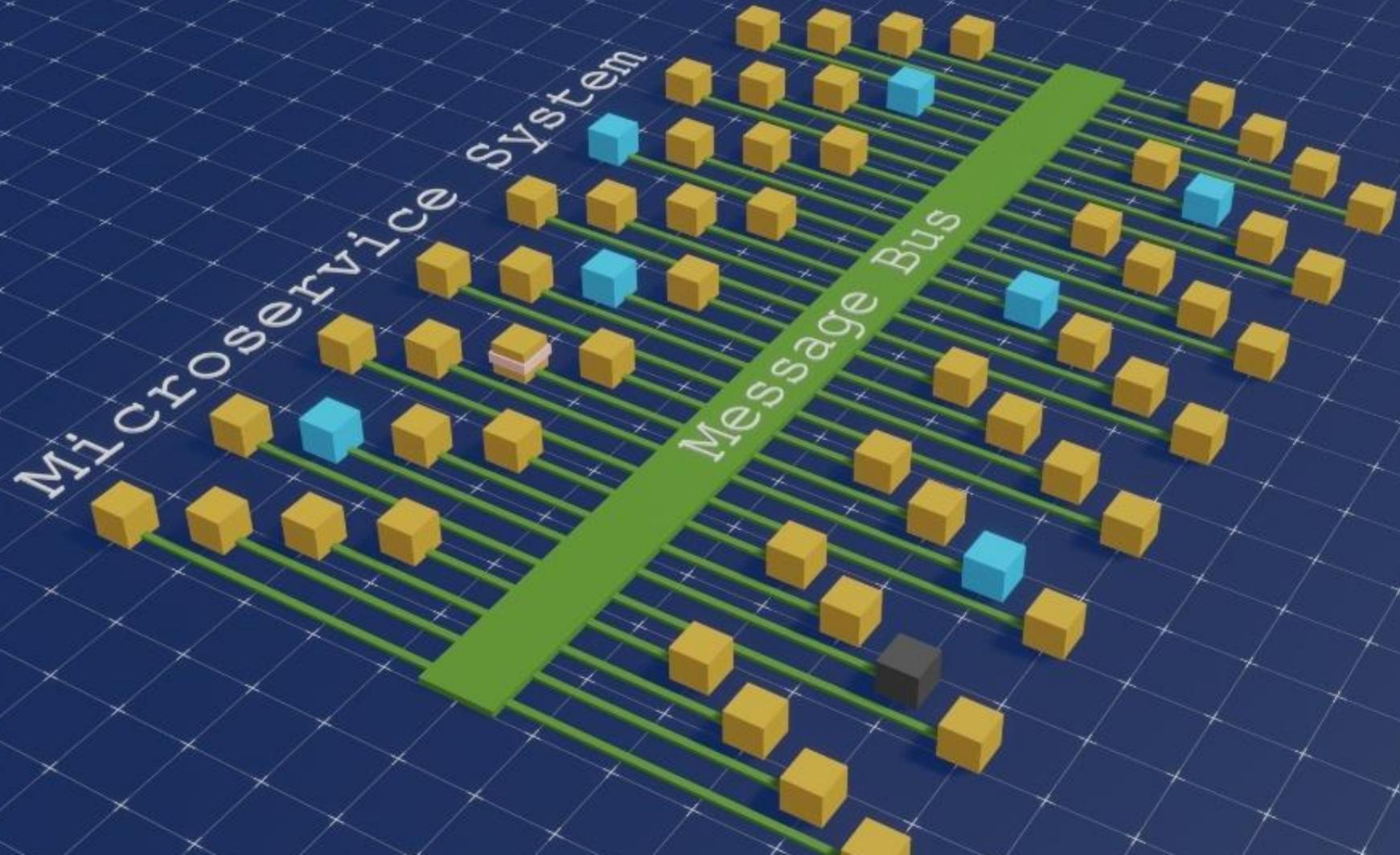


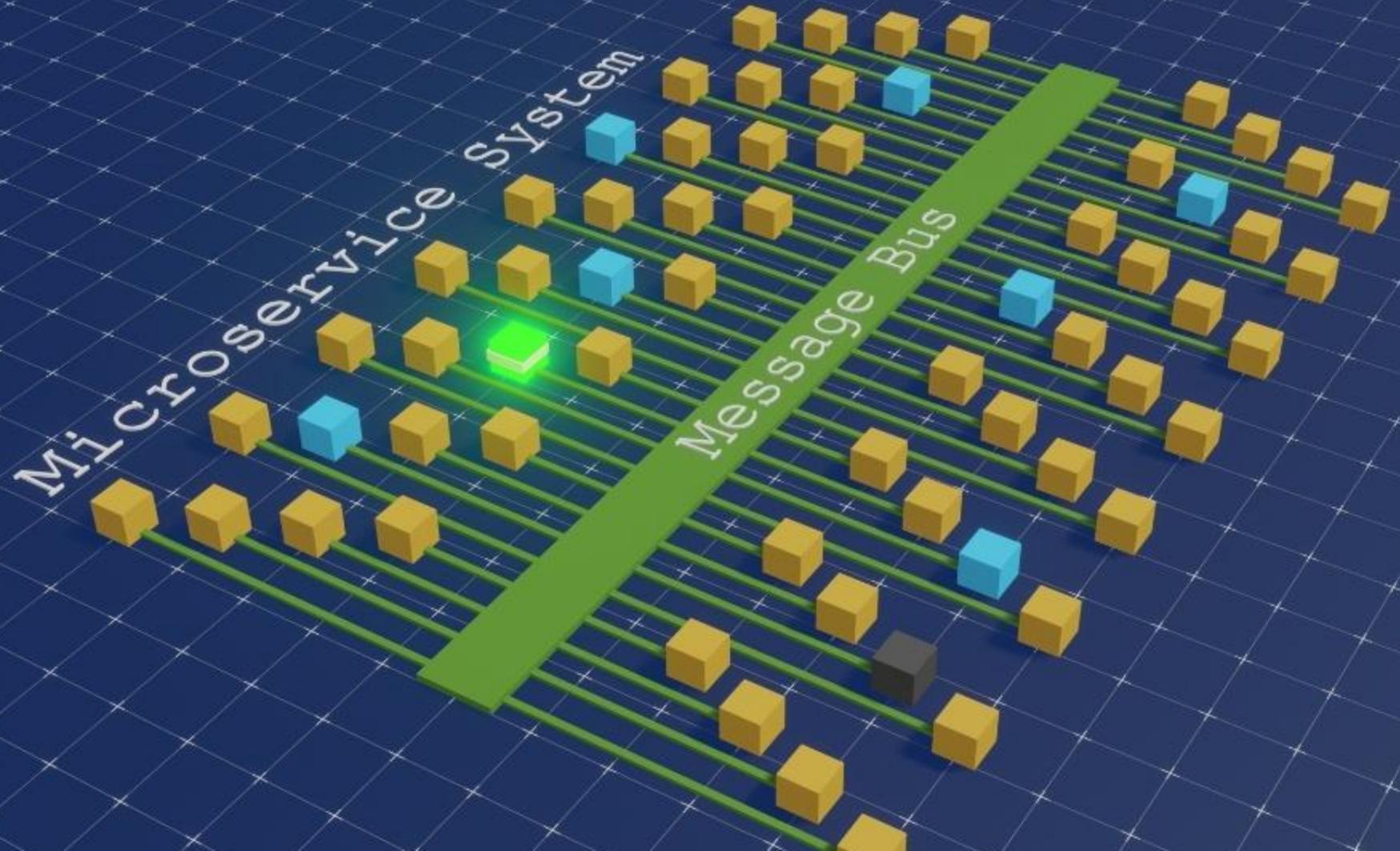


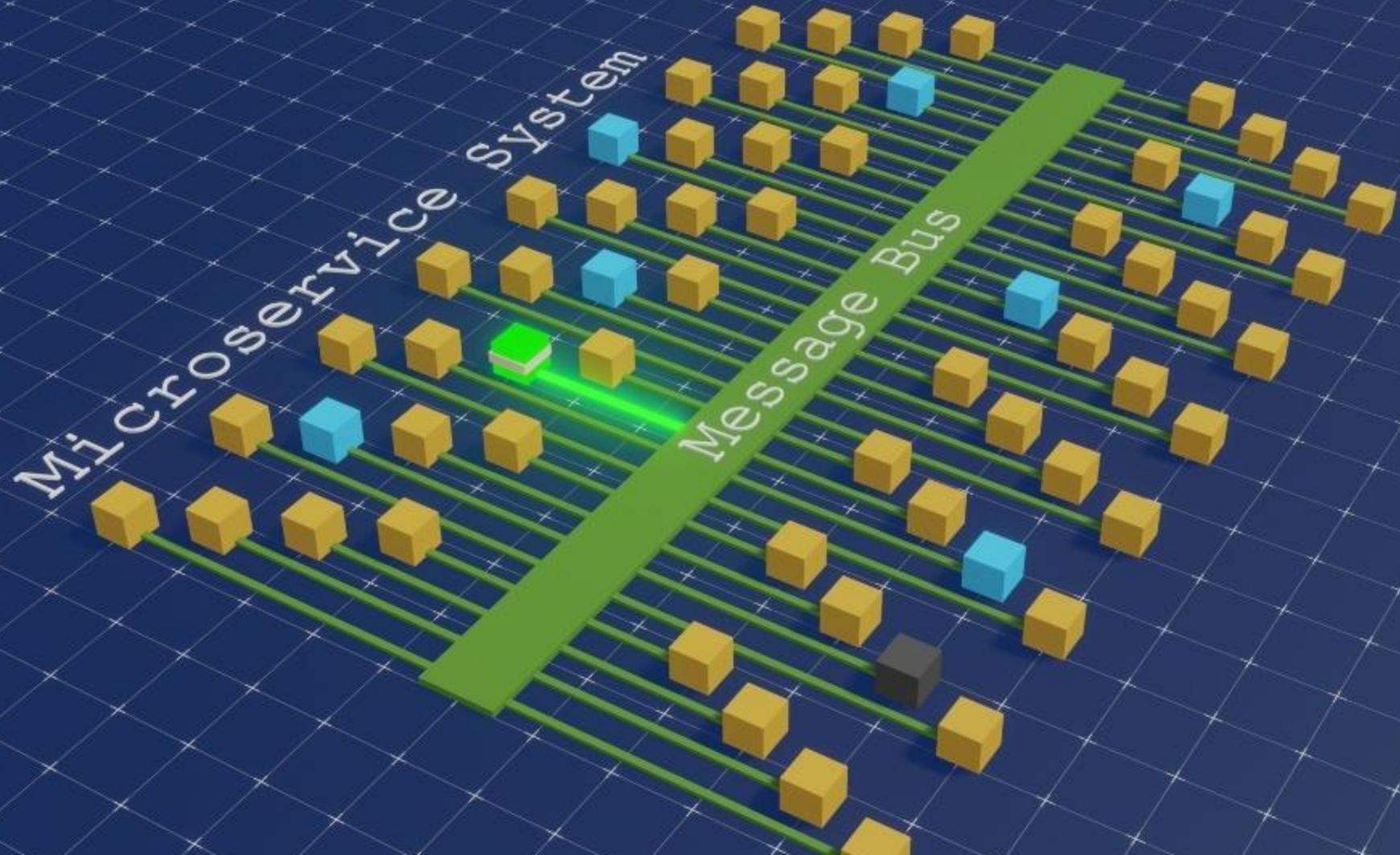


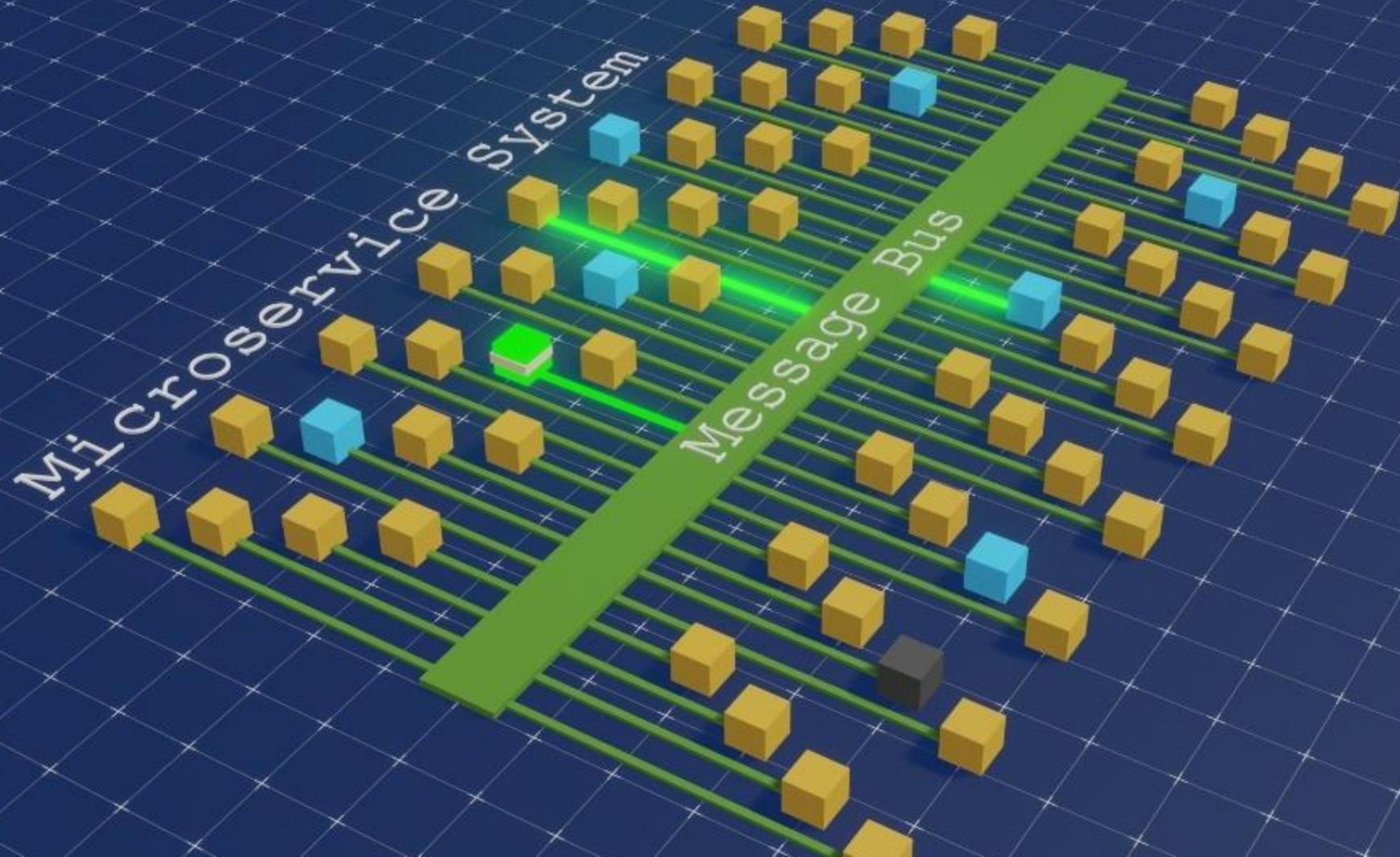


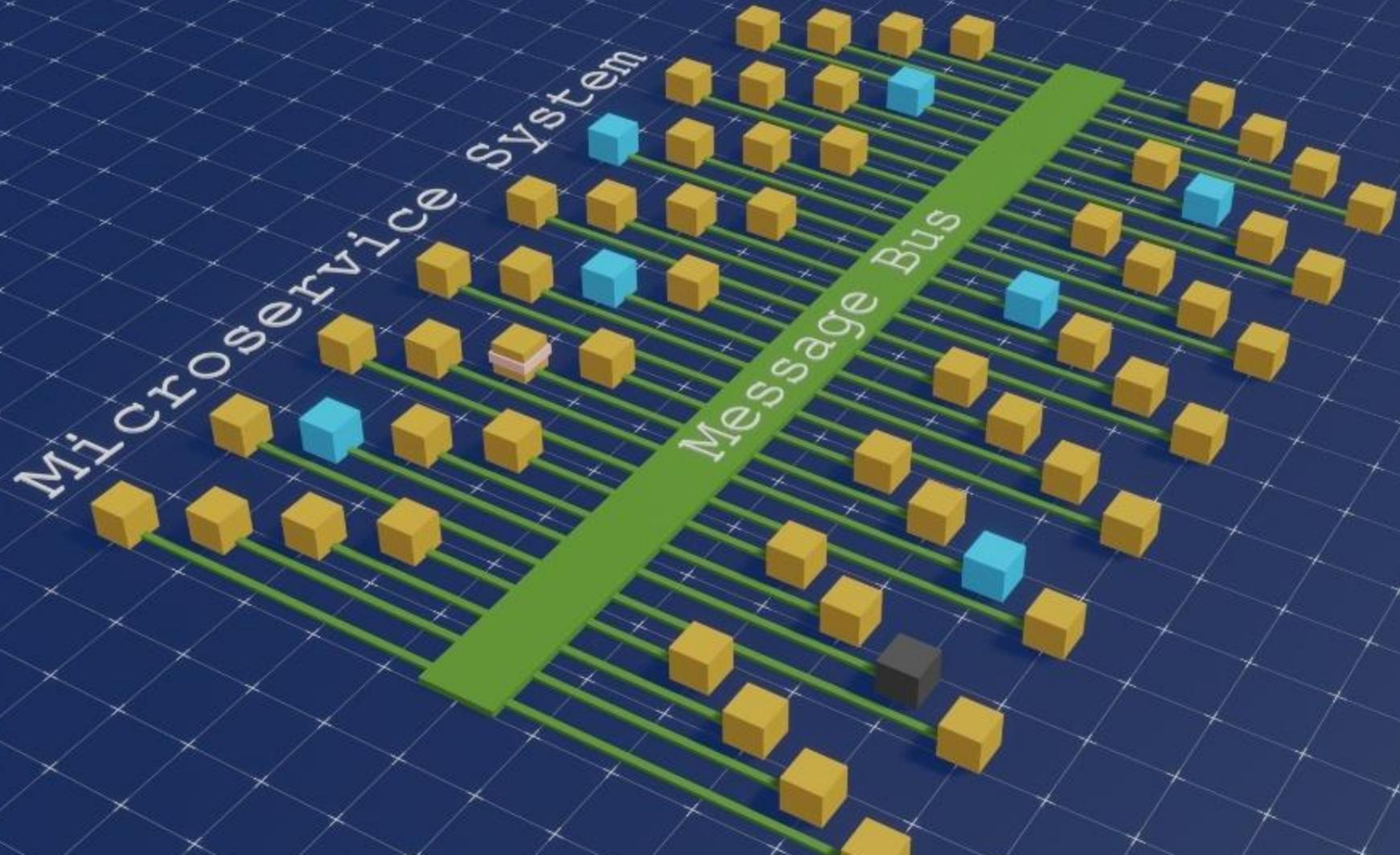


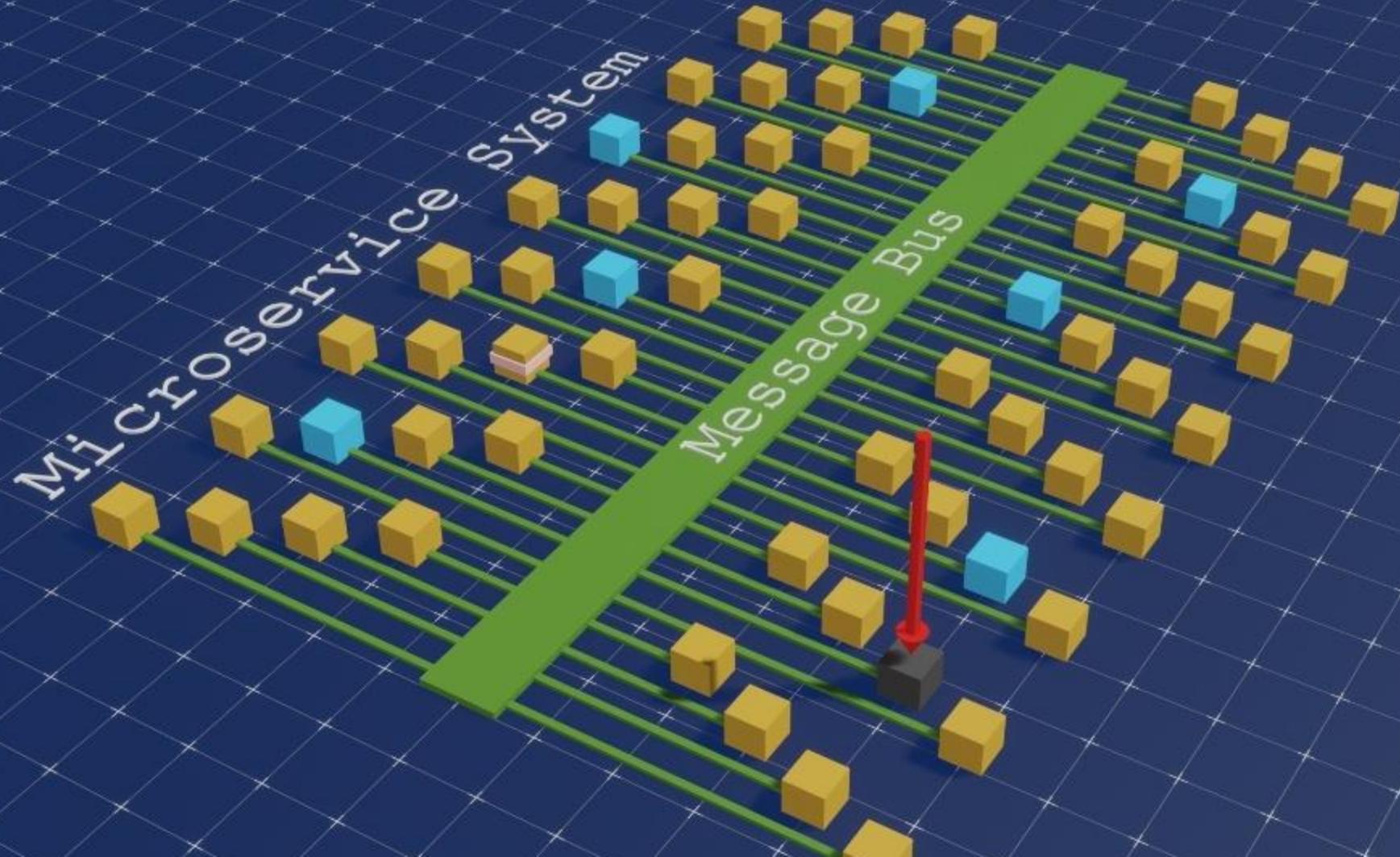


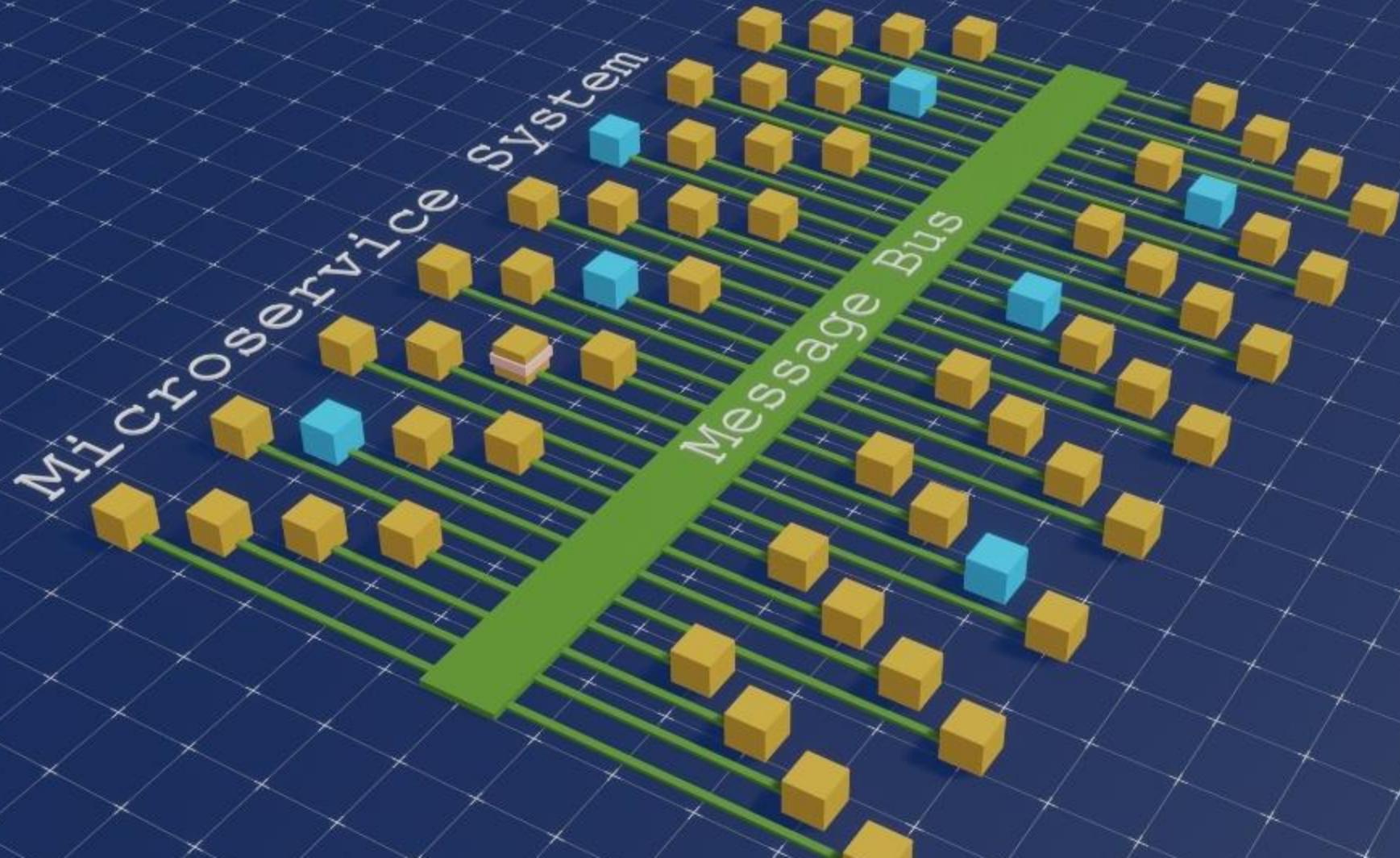


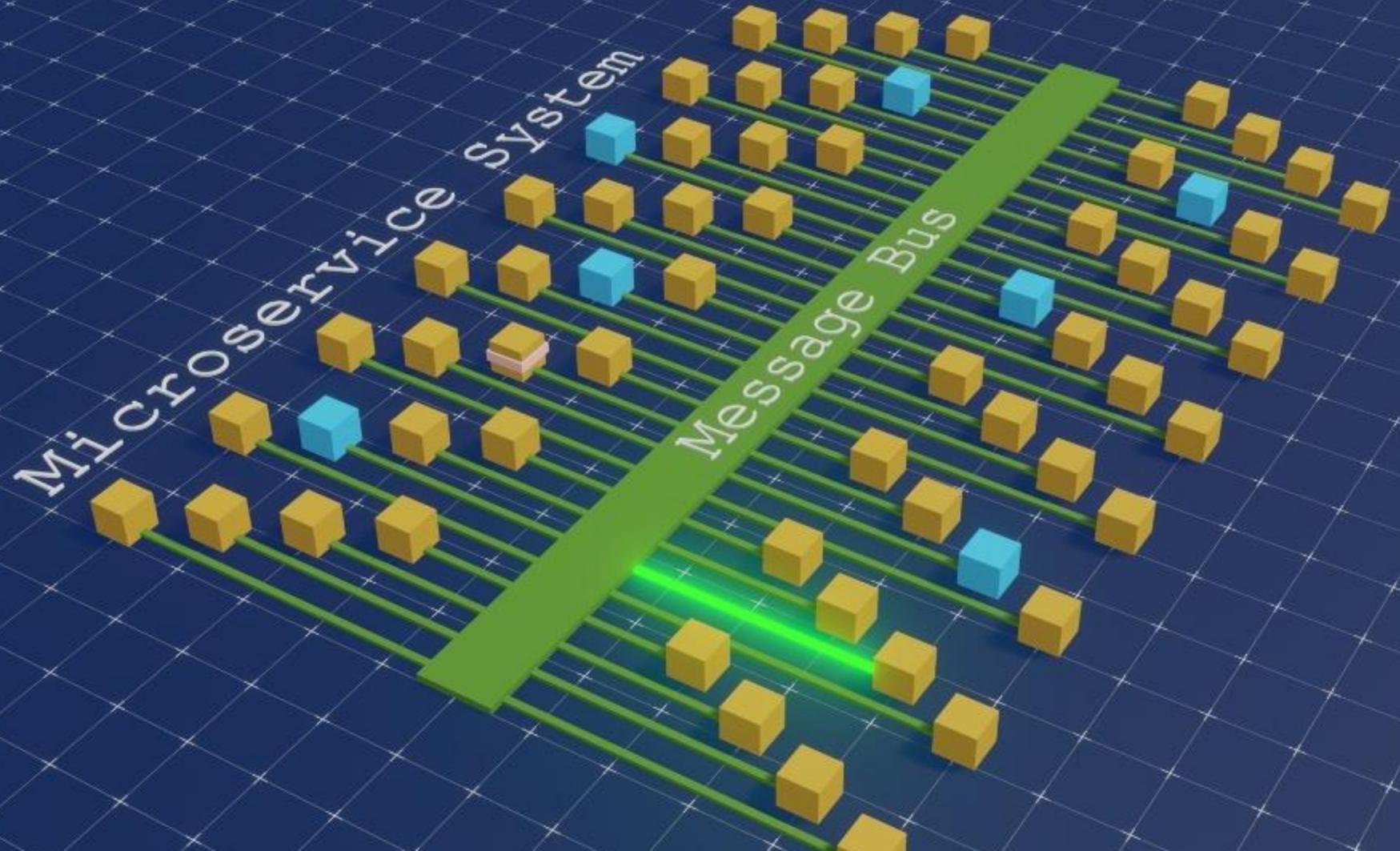


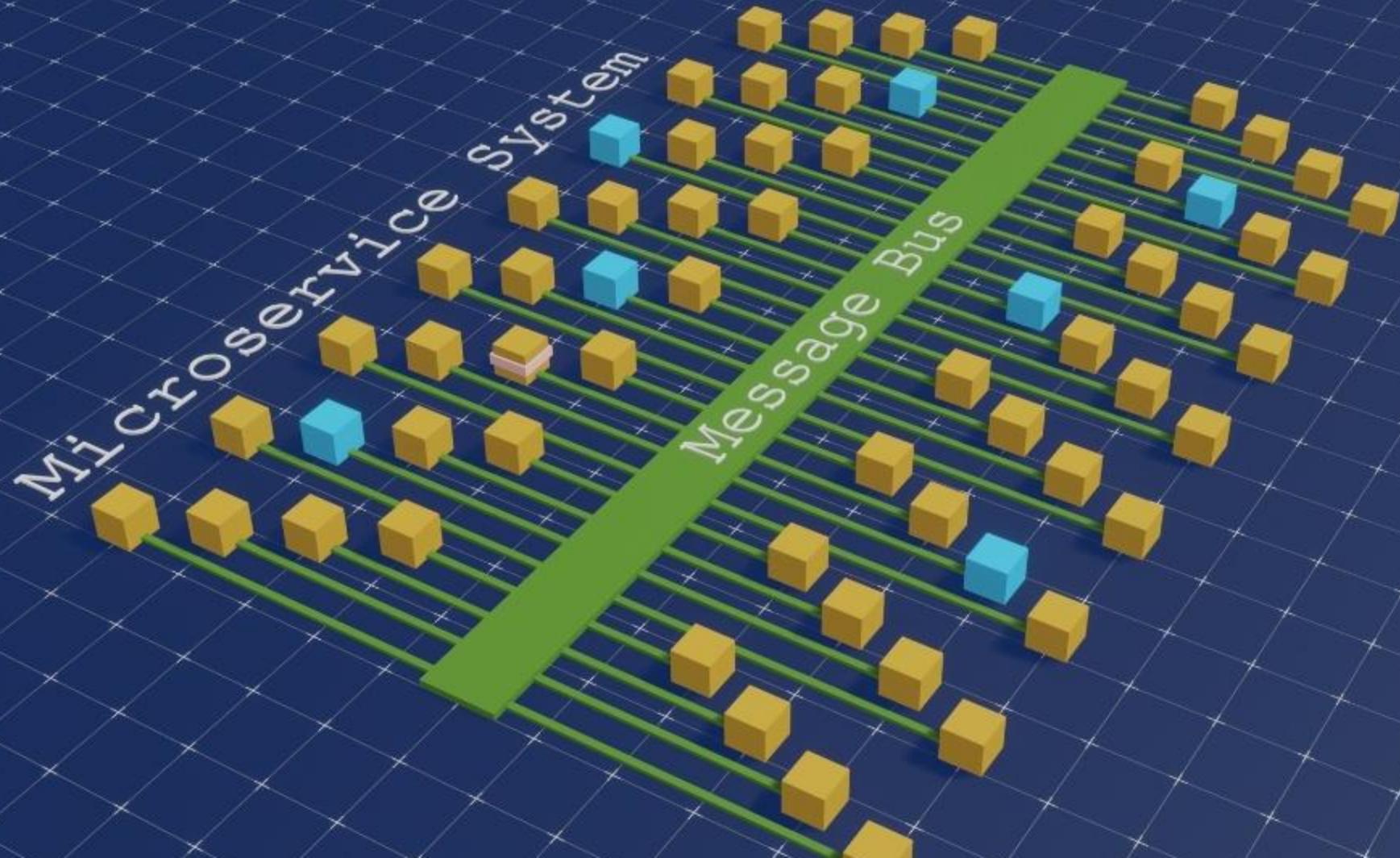


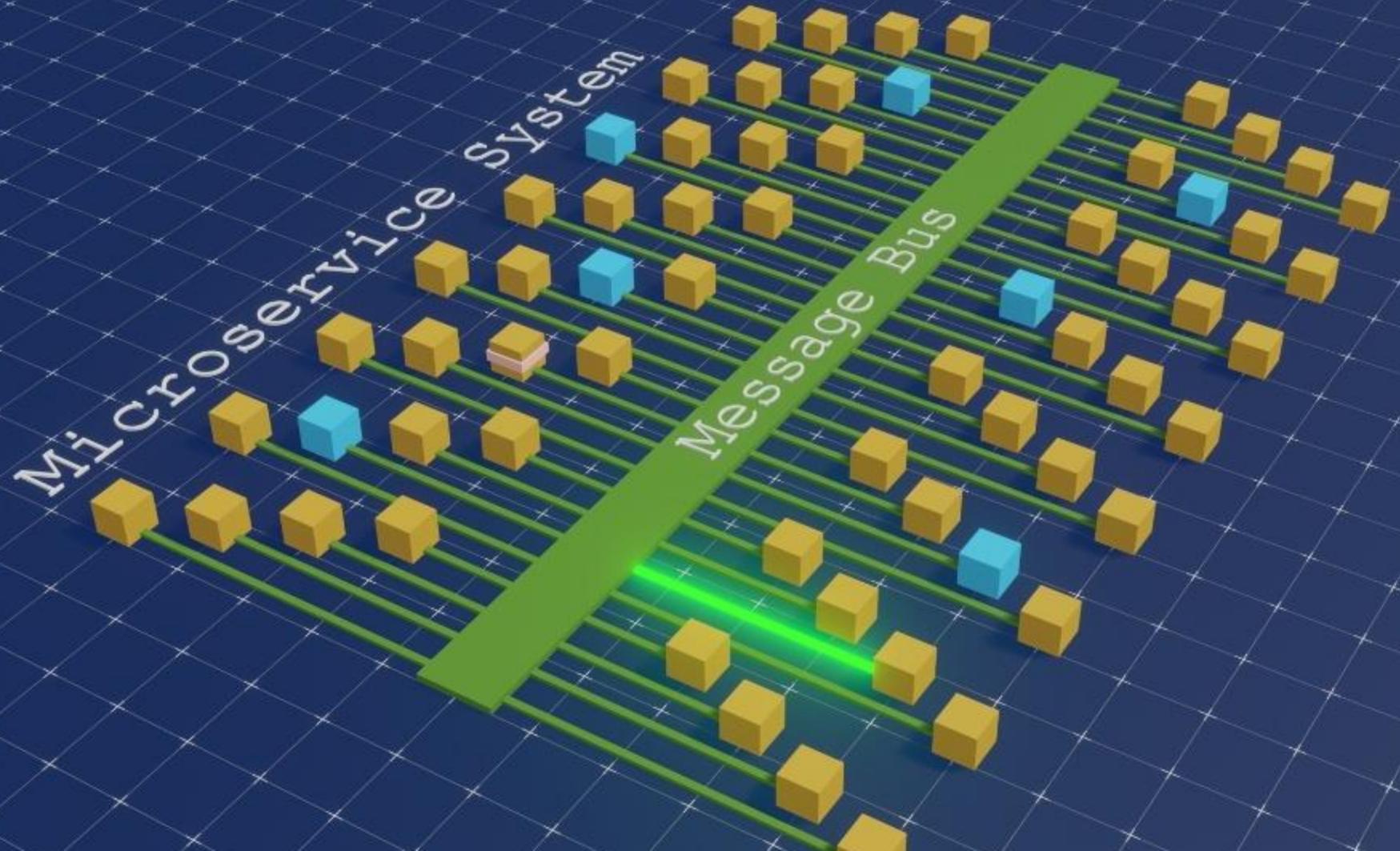


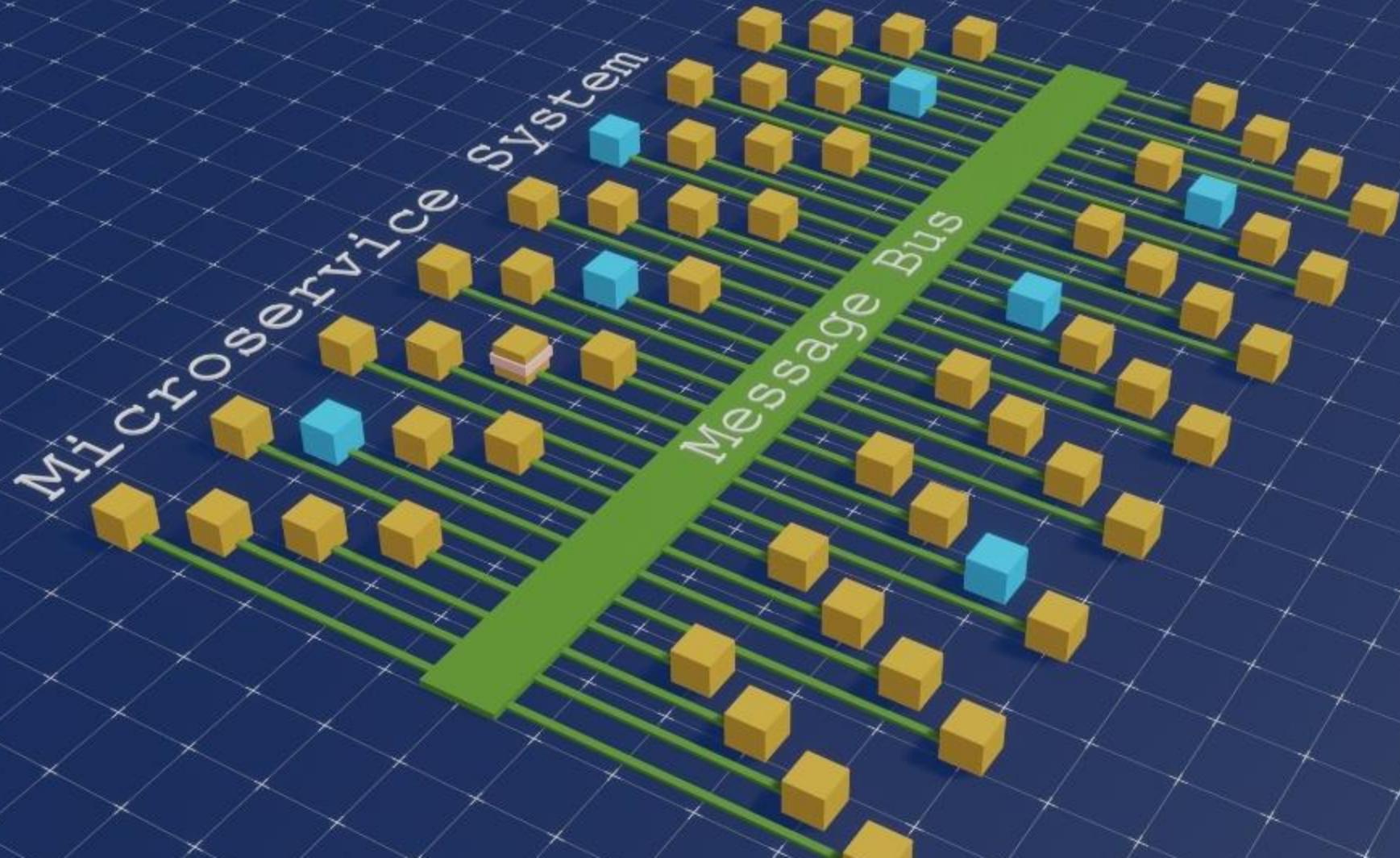


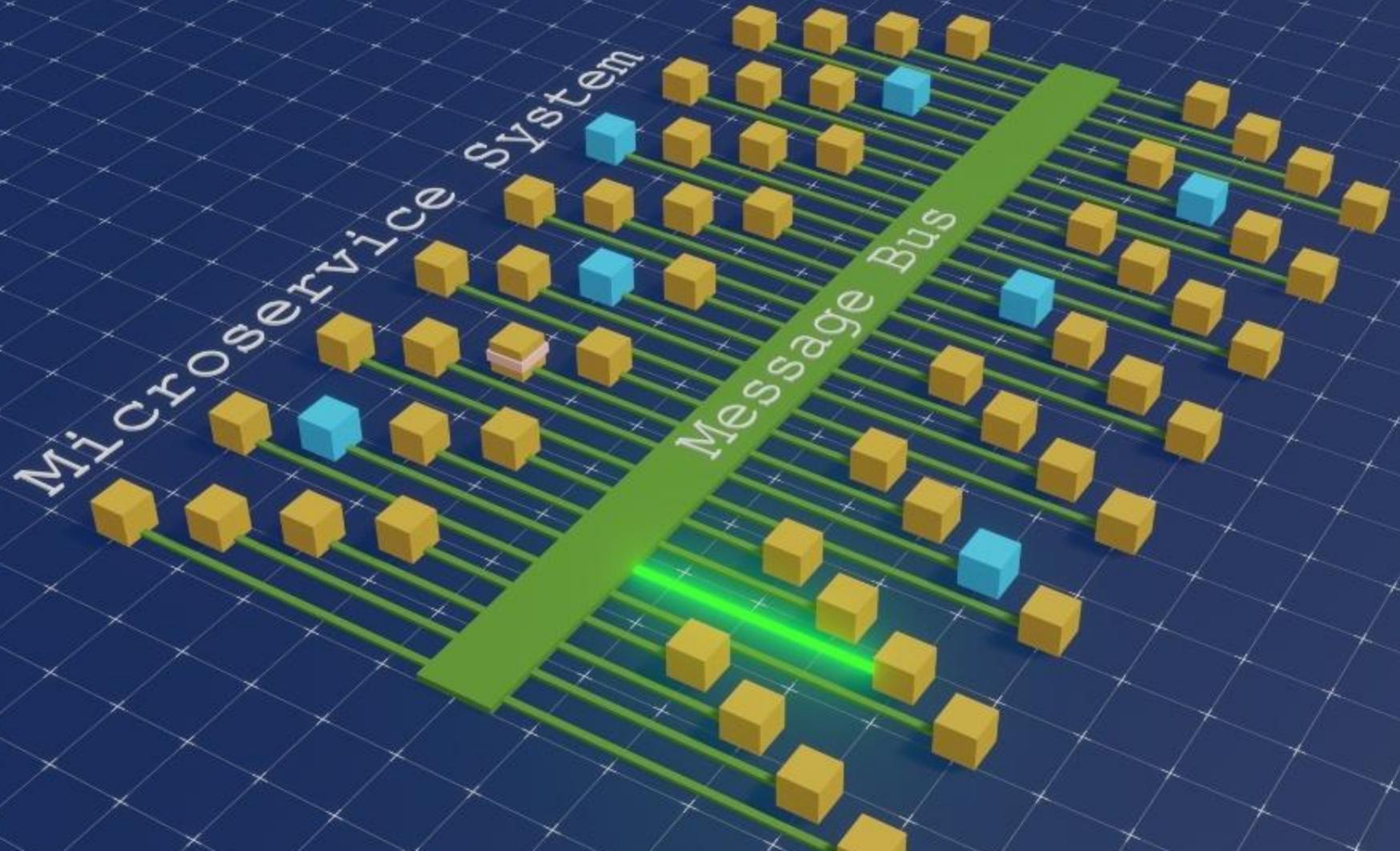


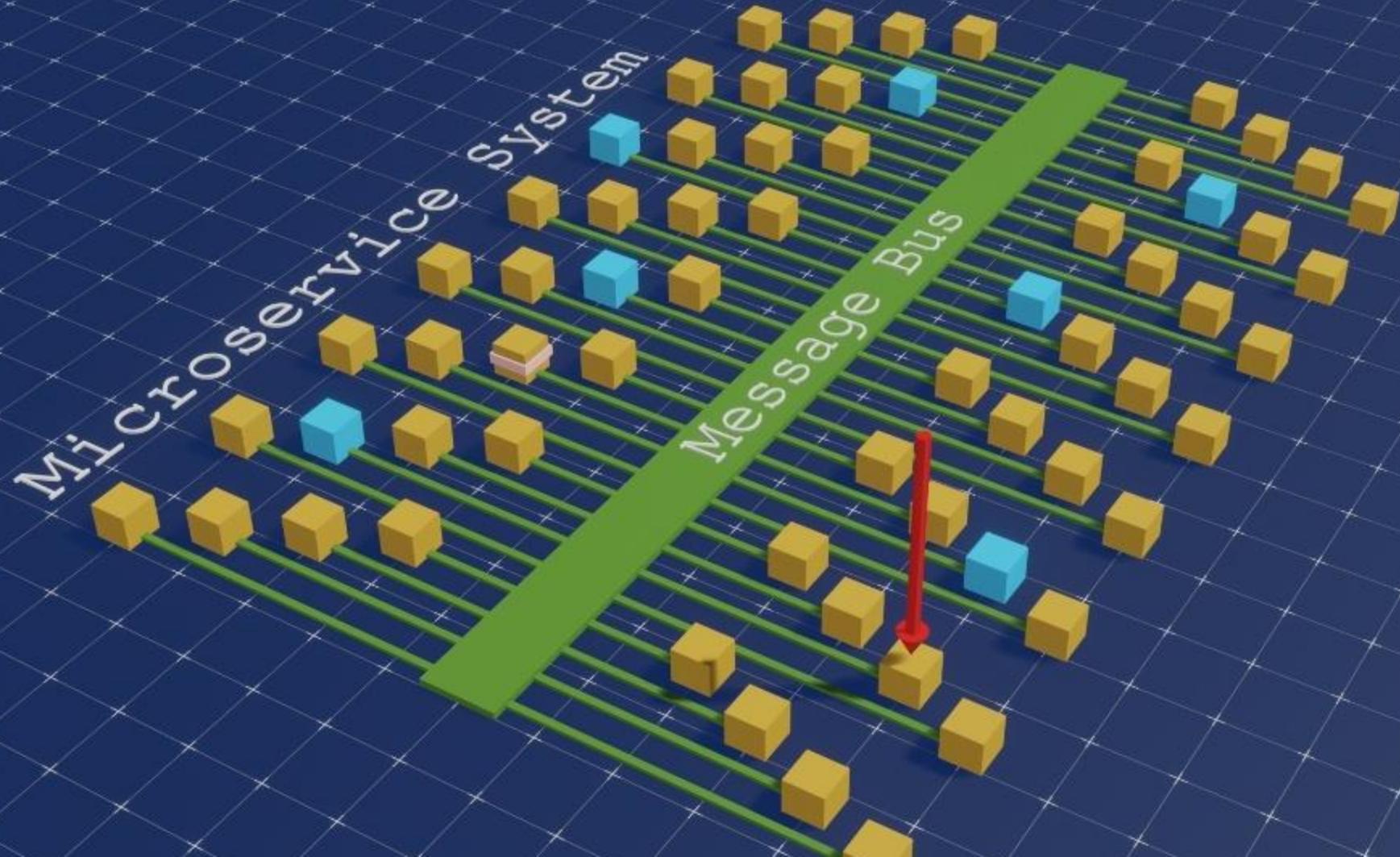












?