



# Unity 云原生 分布式运行时

2023

通往元宇宙的基石



# 概要

- 元宇宙时代传统游戏引擎面临的挑战
- Unity云原生分布式渲染
- Unity云原生分布式计算
- Demo/效果展示（内容未定）
- 产品路线图与未来规划



**“The metaverse is  
the next generation of the Internet  
that is always real-time, mostly 3D,  
mostly interactive, mostly social and  
mostly persistent.”**

— John Riccitiello, CEO, Unity



# 元宇宙的特征

## 规模庞大

巨大、互联的虚拟世界：参与人数多，环境规模大，交互种类复杂多样

## 3D实时交互

沉浸式3D场景，玩家间实时互动，高响应度与互动性

## 持续稳定

长期持续、稳定的虚拟世界，玩家对世界的改变随时间积累保留



# 传统游戏引擎面临的挑战



# 传统游戏引擎面临的挑战

- 巨大的渲染压力
  - 大规模的高清世界
  - 数量庞大的动态元素
- 巨大的计算压力
  - 大量玩家实时交互
  - 超大规模的物理仿真
- 巨大的网络带宽压力
- 可扩展性、可伸缩性





# Unity云原生 分布式运行时



**Unity**云原生  
分布式渲染

+

**Unity**云原生  
分布式计算



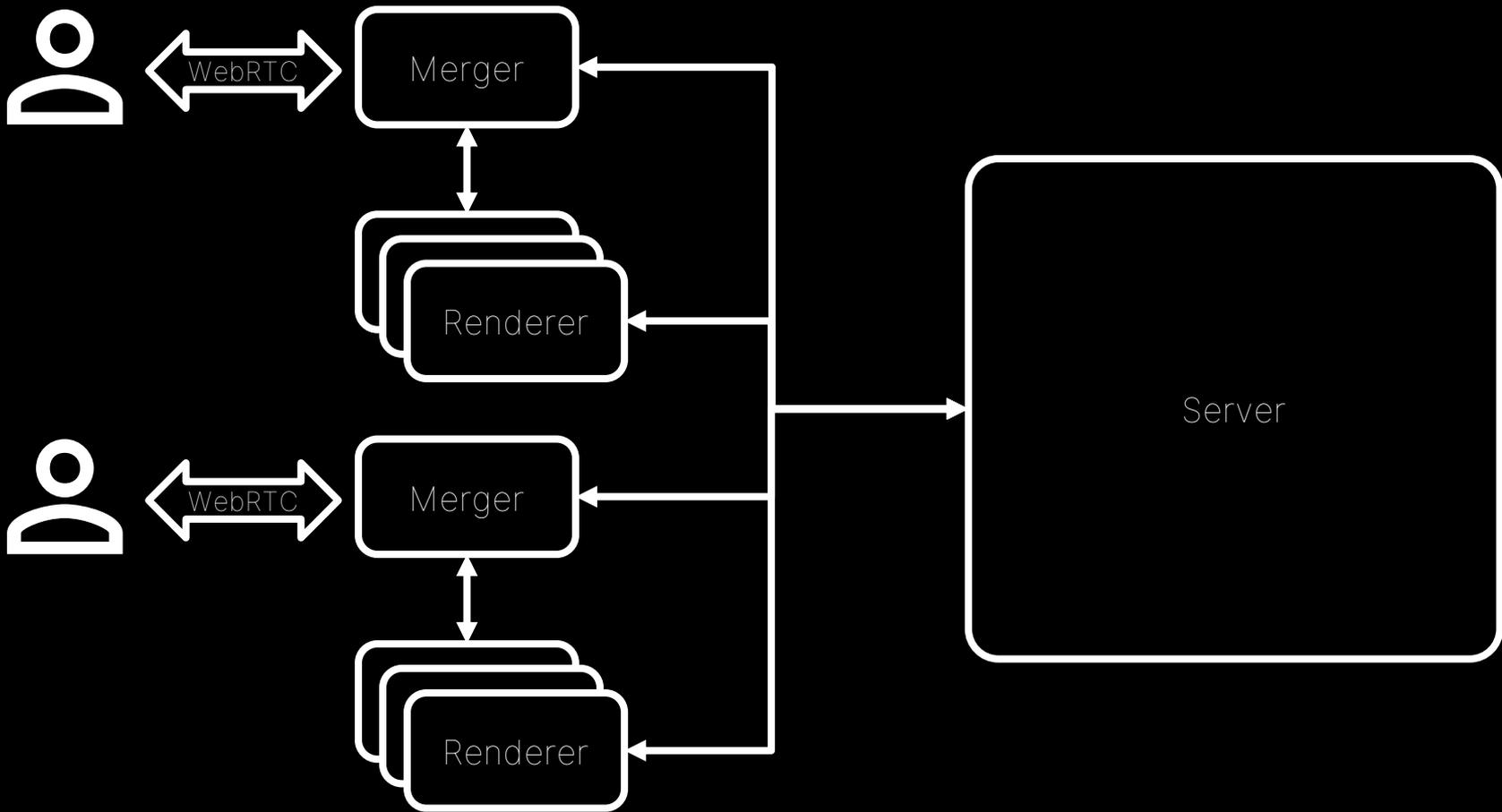
# 为何选择DOTS

- 高性能
- 存算分离
- 适合并行



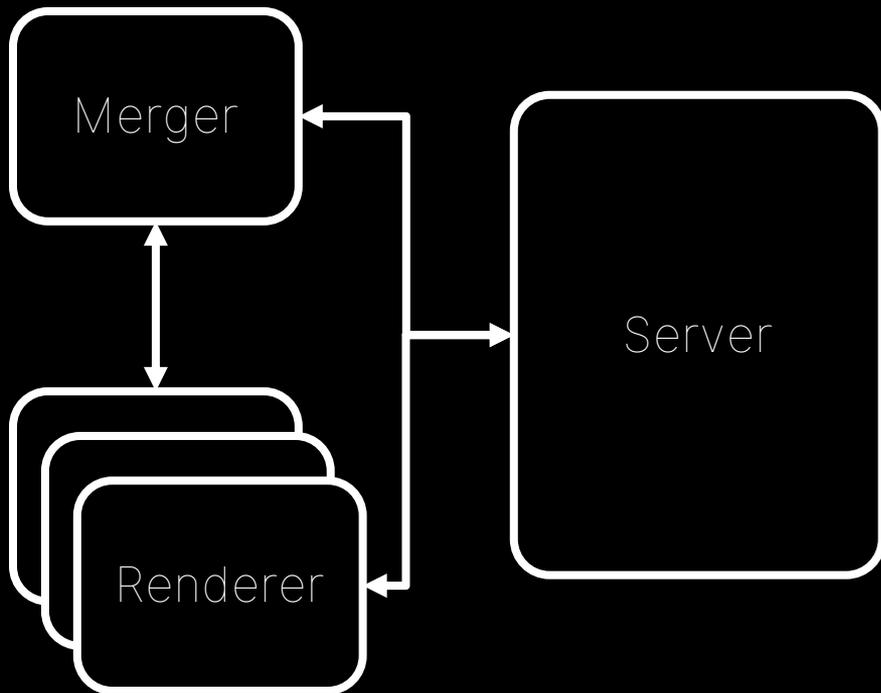


# Unity云原生分布式渲染



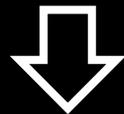
# 分布式渲染架构

- Merger
  - 合并Renderer的画面
  - 接收、转发用户输入
  - 向客户端推流
- Renderer
  - 渲染游戏世界
- Server
  - 处理逻辑计算、更新游戏世界
  - 向Render Group同步世界数据
  - 管理Render Group更新频率



# 拆分渲染任务

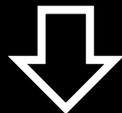
→ 根据屏幕空间拆分





## 拆分渲染任务

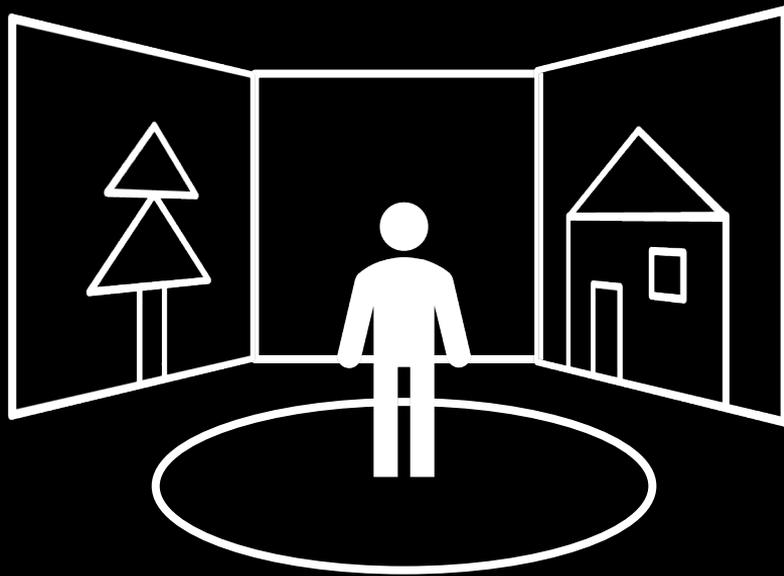
- 根据屏幕空间拆分
- 根据时间拆分





## 拆分渲染任务

- 根据屏幕空间拆分
- 根据时间拆分
- 根据场景拆分



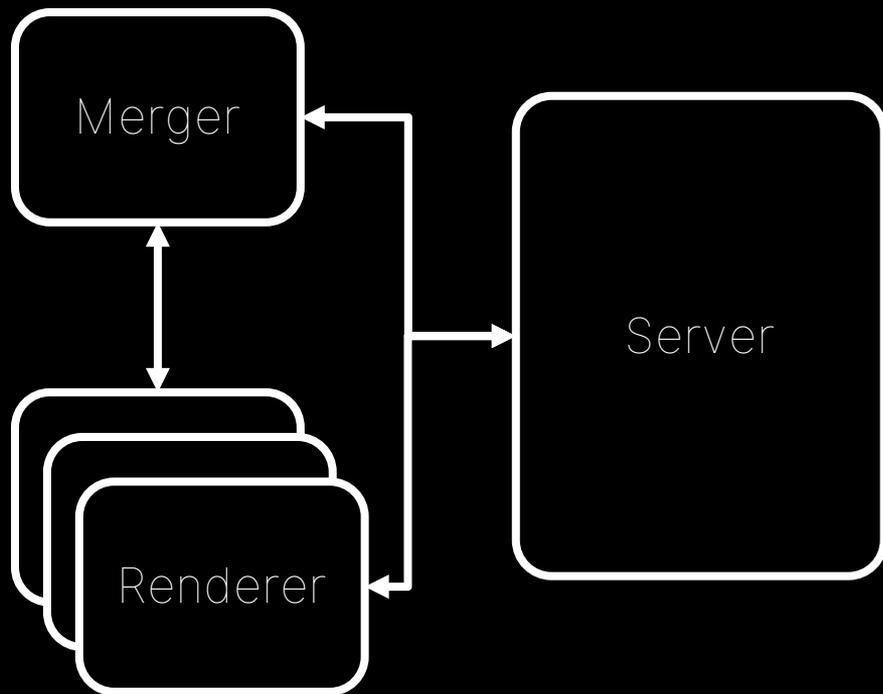


# 拆分渲染任务

- 根据屏幕空间拆分
- 根据时间拆分
- 根据场景拆分
- .....

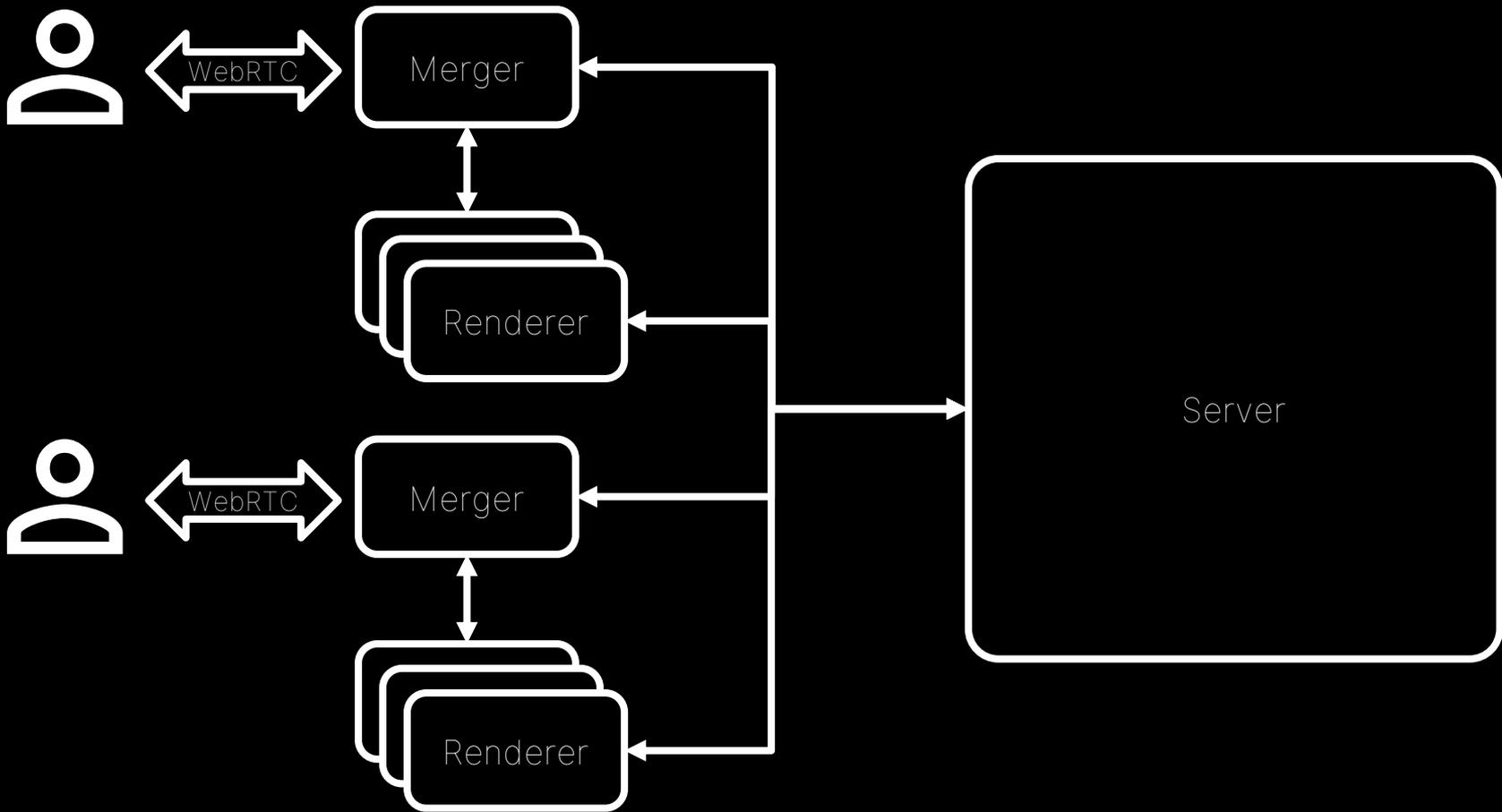
# 分布式渲染能力

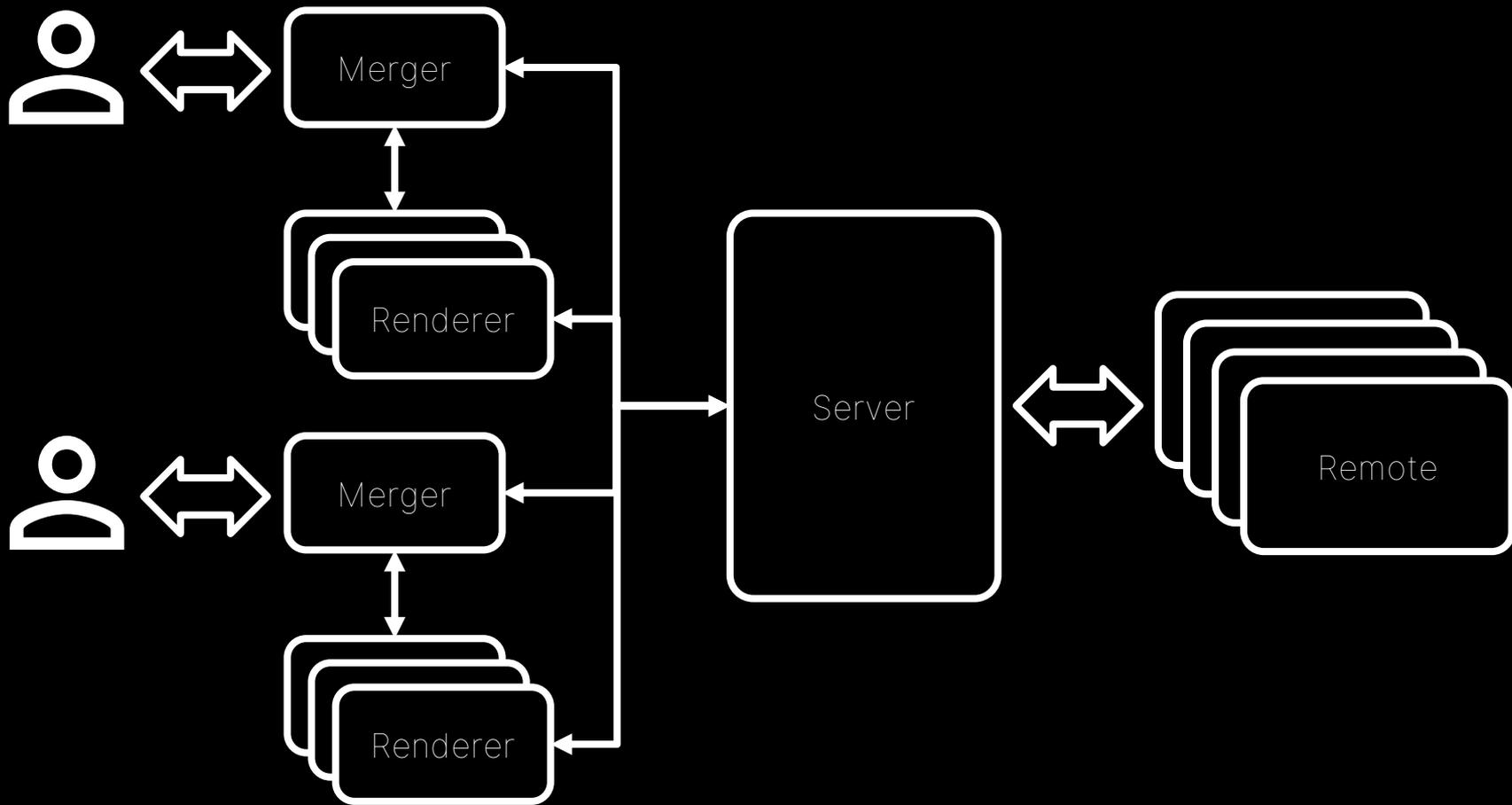
- 统一的渲染调度
- 一致的世界数据
- 高效的画面传输





# Unity云原生分布式计算





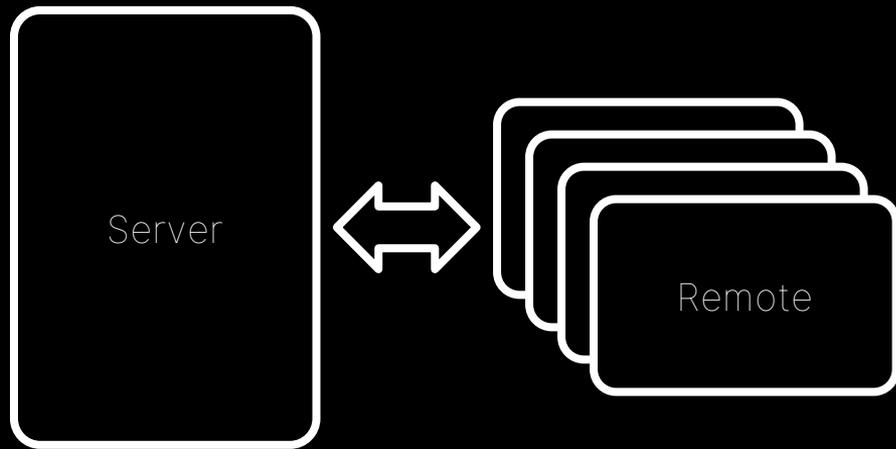
# 分布式计算架构

## → Server

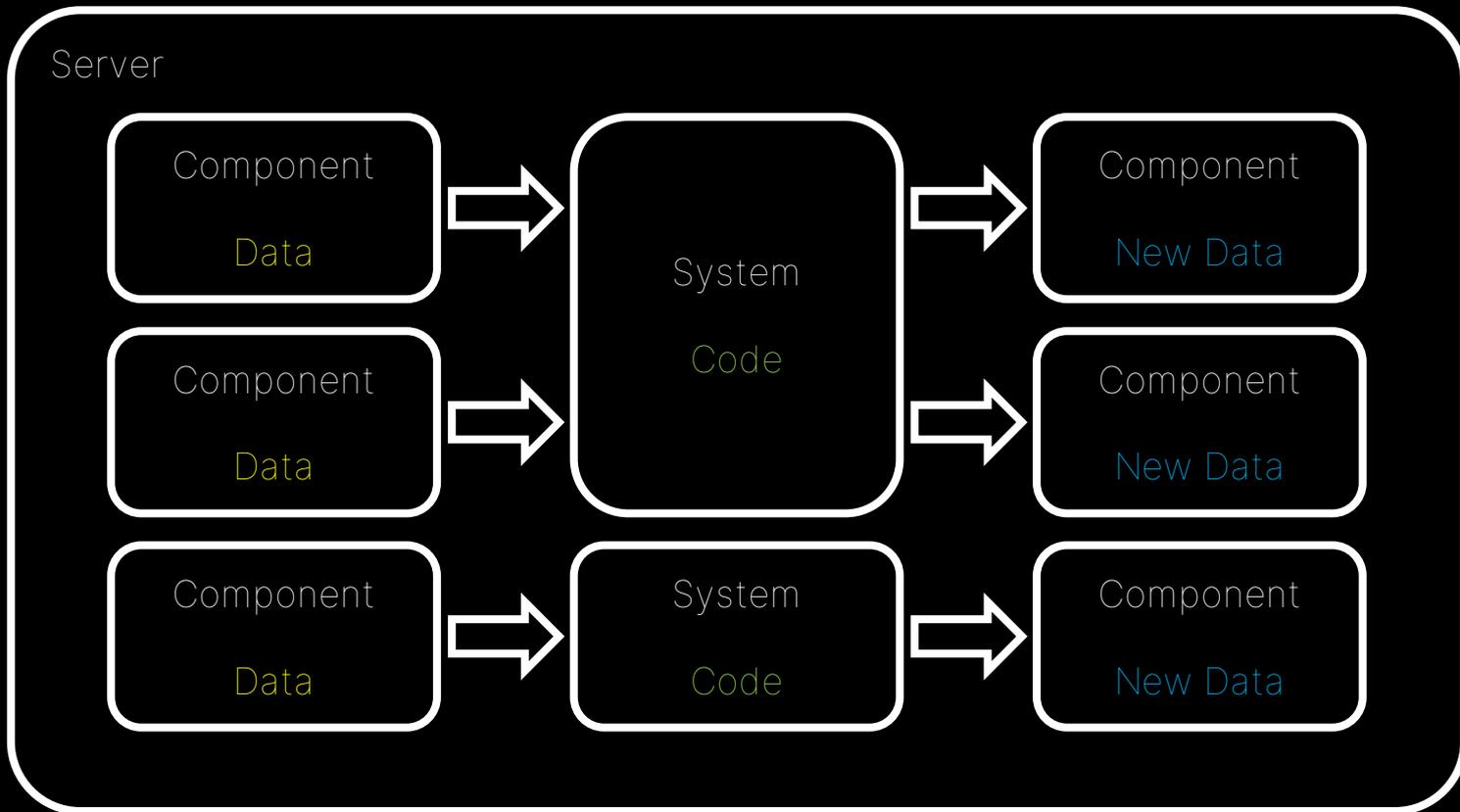
- 向Remote分发计算任务、数据
- 从Remote获取计算结果
- 更新世界数据

## → Remote

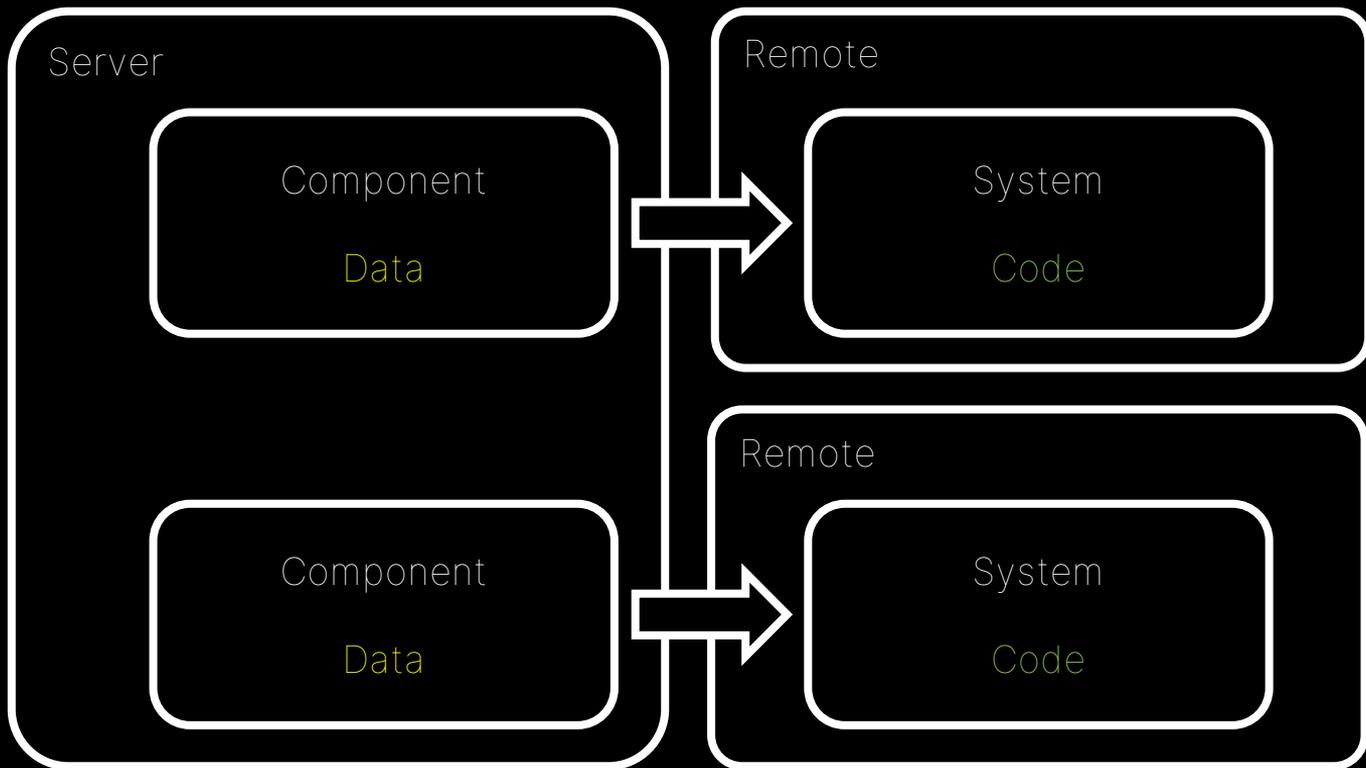
- 从Server获取计算任务、数据
- 进行逻辑计算，返回计算结果
- 动态扩容



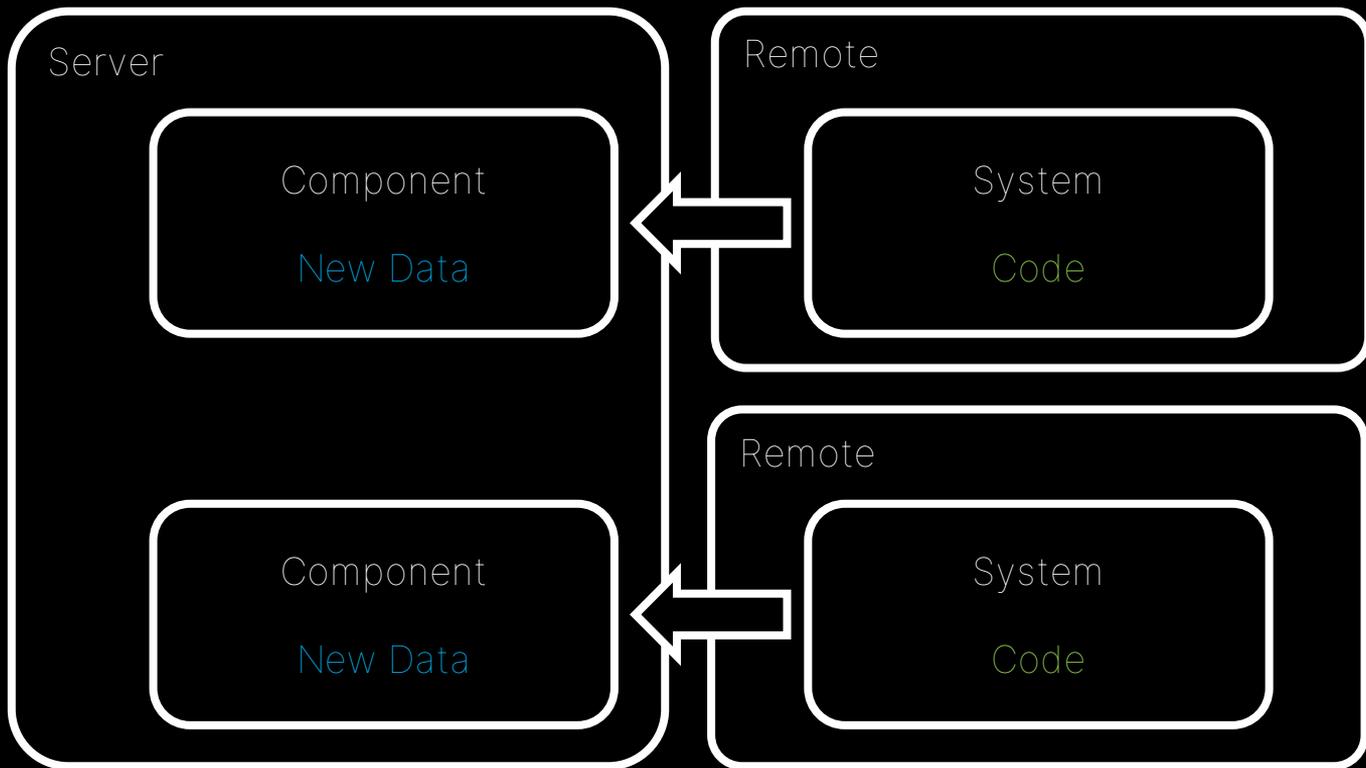
# 分配计算任务



# 分配计算任务



# 分配计算任务



# 分配计算任务

V Πύτ Άρή üčÄřřΠ, ř' öÄŦ

- VI
- VII
- VIII
- IX
- X
- XI
- XII
- L
- VIV
- VV
- VVI
- VVII
- VVIII
- VIX
- VX
- VXI
- VXII
- VL
- VIV
- VV
- VVI
- VVII
- VVIII
- VIX
- VX
- VXI
- VXII
- VL
- VIV



# DEMO/效果展示



青霄？  
星际广场/世界杯？  
**In-House Demo?**



# 产品路线图与未来规划



# Roadmap

→ Roadmap Here



# 未来规划

→ GameObject/MonoBehaviour?

→ 端云协同？

→ 异构计算 ( x86/ARM ) ？

→ 跨渲染管线 ( HDRP/URP ) ？

→ 边缘计算？

→ .....



**Thank you**

2023

