

PHANTOM LIBERTY

GDC

MARCH 18-22, 2024
SAN FRANCISCO, CA

A close-up, high-quality render of the character V from the video game Cyberpunk 2077. V is a woman with long, vibrant red hair, looking intensely at the viewer. She has a cybernetic eye and a small, glowing blue display on her cheek. She is holding a red and silver cybernetic pistol. The lighting is dramatic, with strong red and yellow highlights against a dark background.

Anatomy of a frame in CYBERPUNK 2077: PHANTOM LIBERTY

by Charles Tremblay | **VP of Technology**

Why this talk?

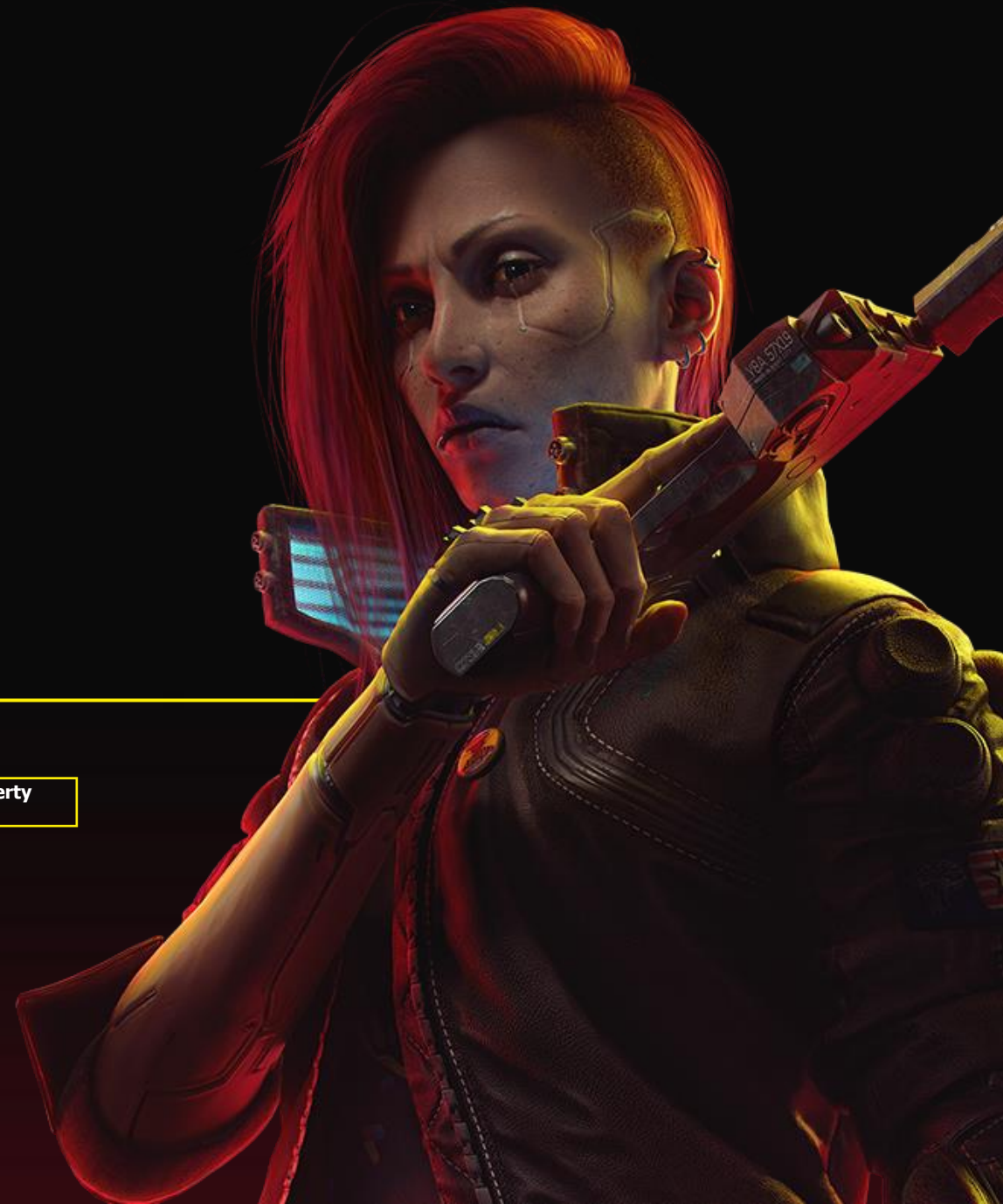
Answering one of the most asked question from GDC 2023 talk

- How did you multithread systems ?

Achieving great scalability on CPU is hard!



Road to Cyberpunk 2077 Phantom Liberty





その目で未来を
空きPOD有り

KIROSHI アイ・プラント

DTK

Love

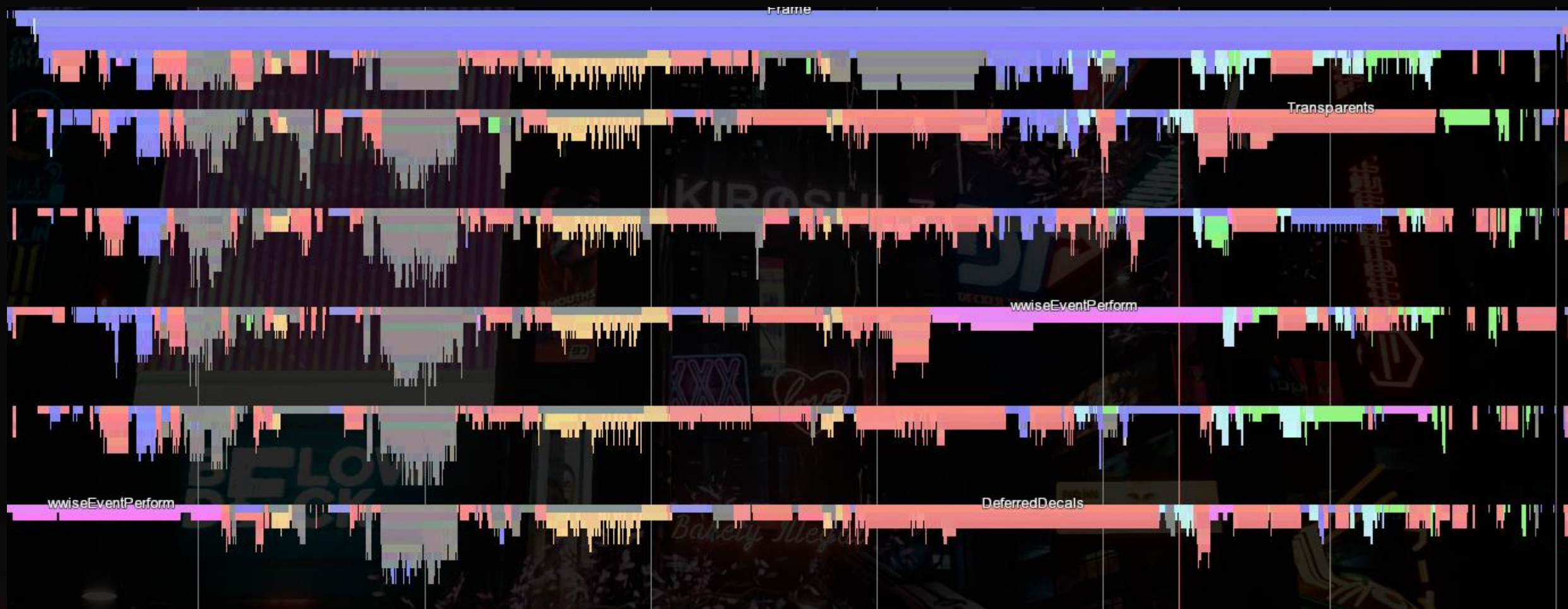
0テル

Barely, Illegal

BELOW
DECK

24/7
STORE

08



Methodology

All numbers were taken on PC

- 13th Gen Intel^(R) Core^(TM) i9-13950HX 2.20 GHz
- Nvidia RTX 4090 Laptop GPU

Number of Job Workers limited to 5

Measurement were made using internal profiler code

Job chain visualization were custom made



Agenda

 Frame Structure

 World Streaming

 Gameplay

 Animation

 Physics

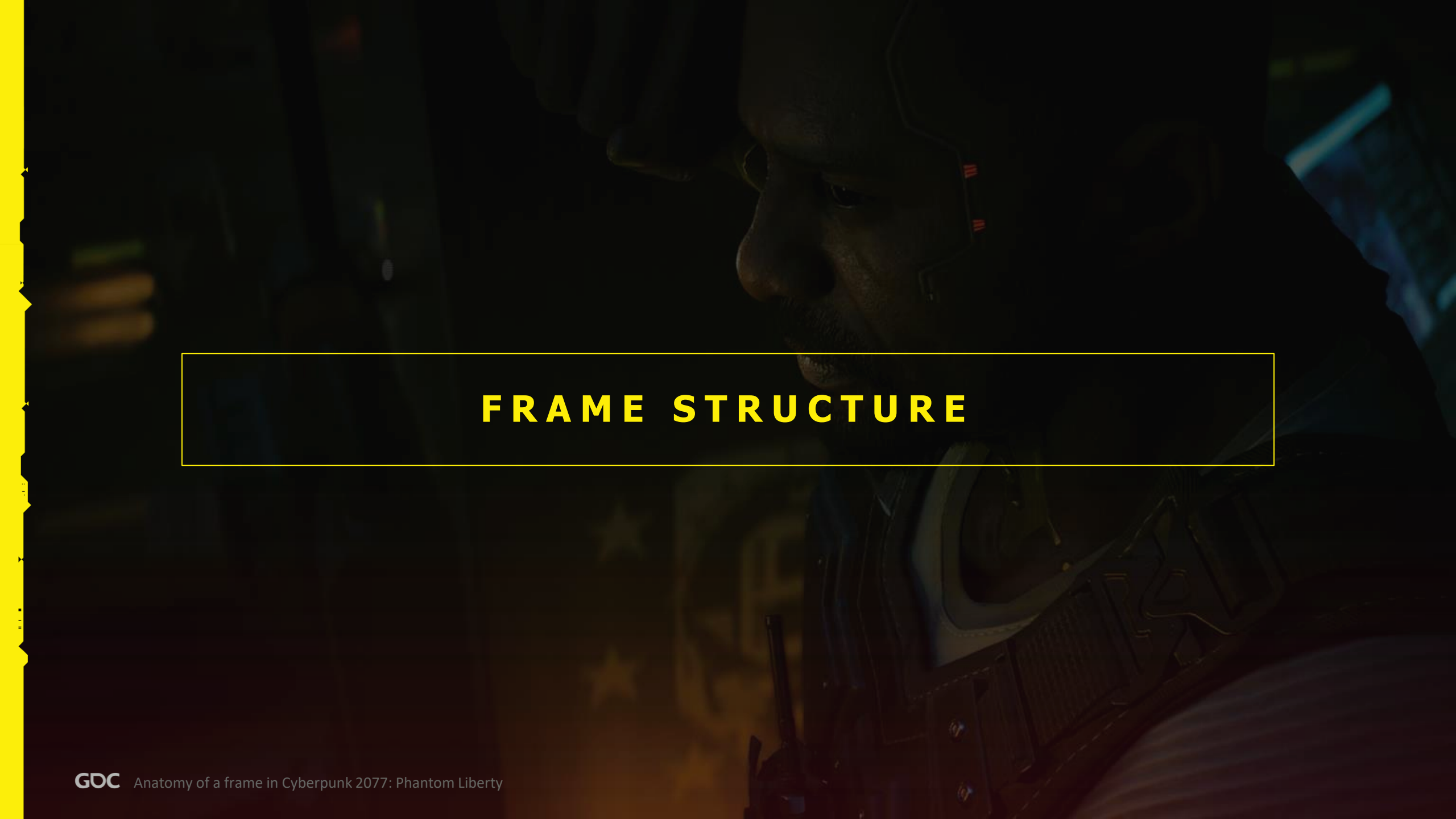
 Audio

 Graphics

 Performance & Scalability

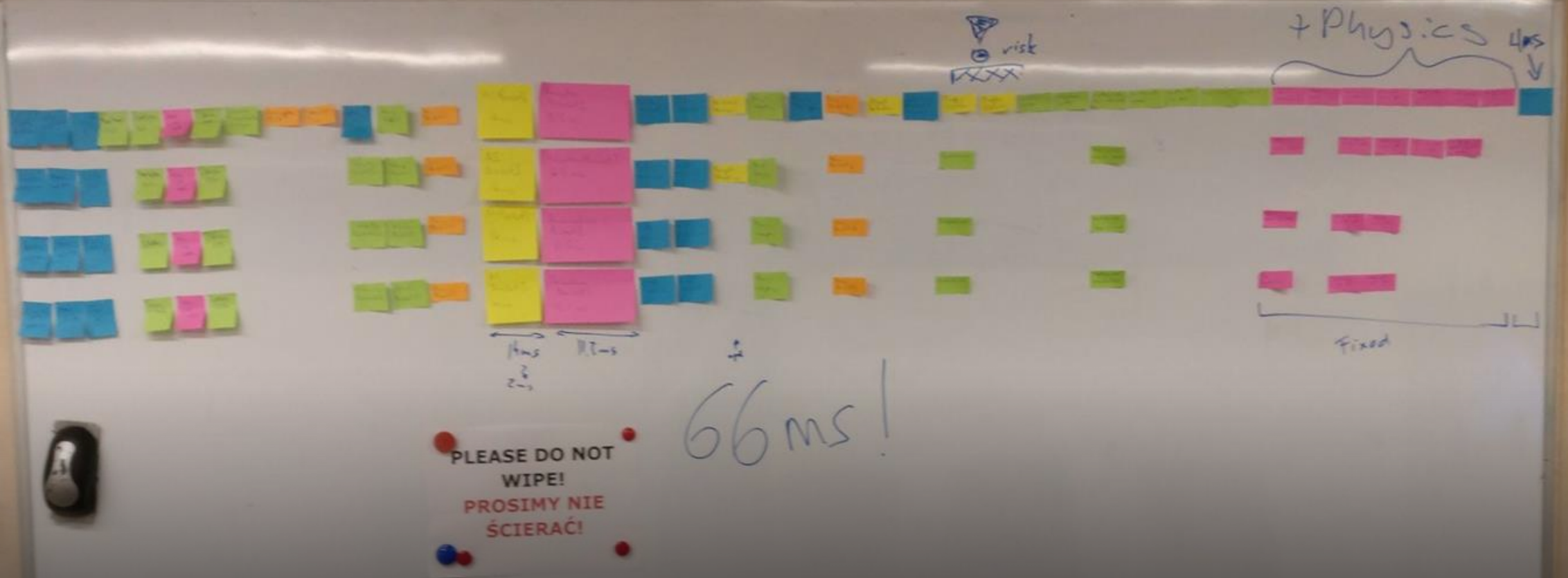
 Conclusion

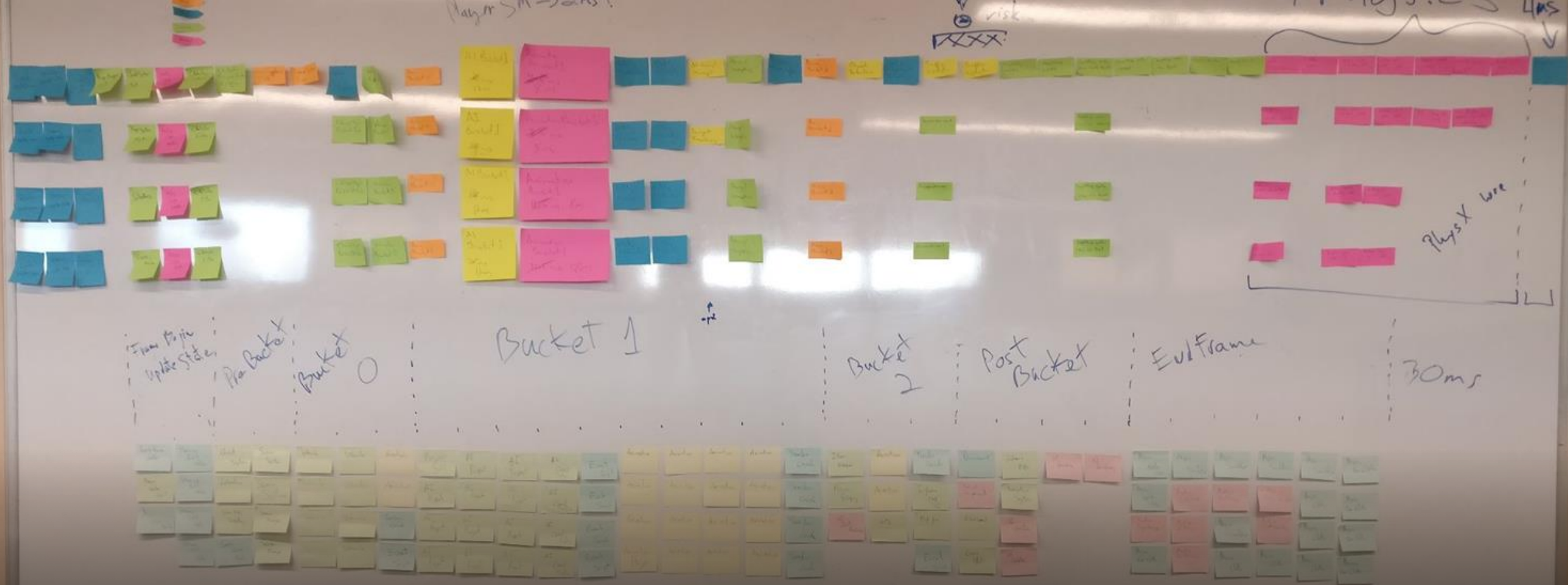




FRAME STRUCTURE

Frame Structure – First Iteration



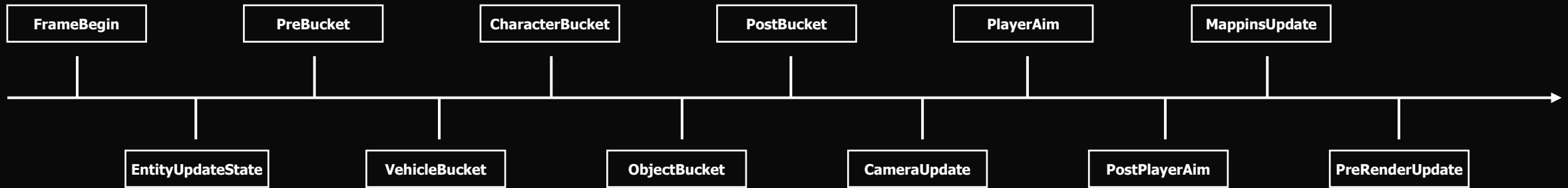


Frame Structure

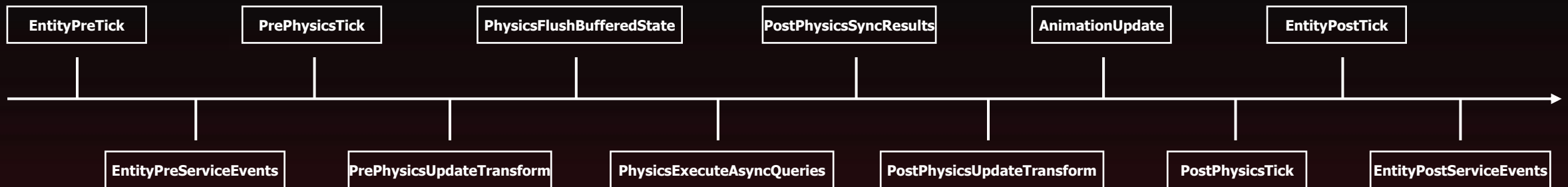
– Second Iteration

Frame Structure

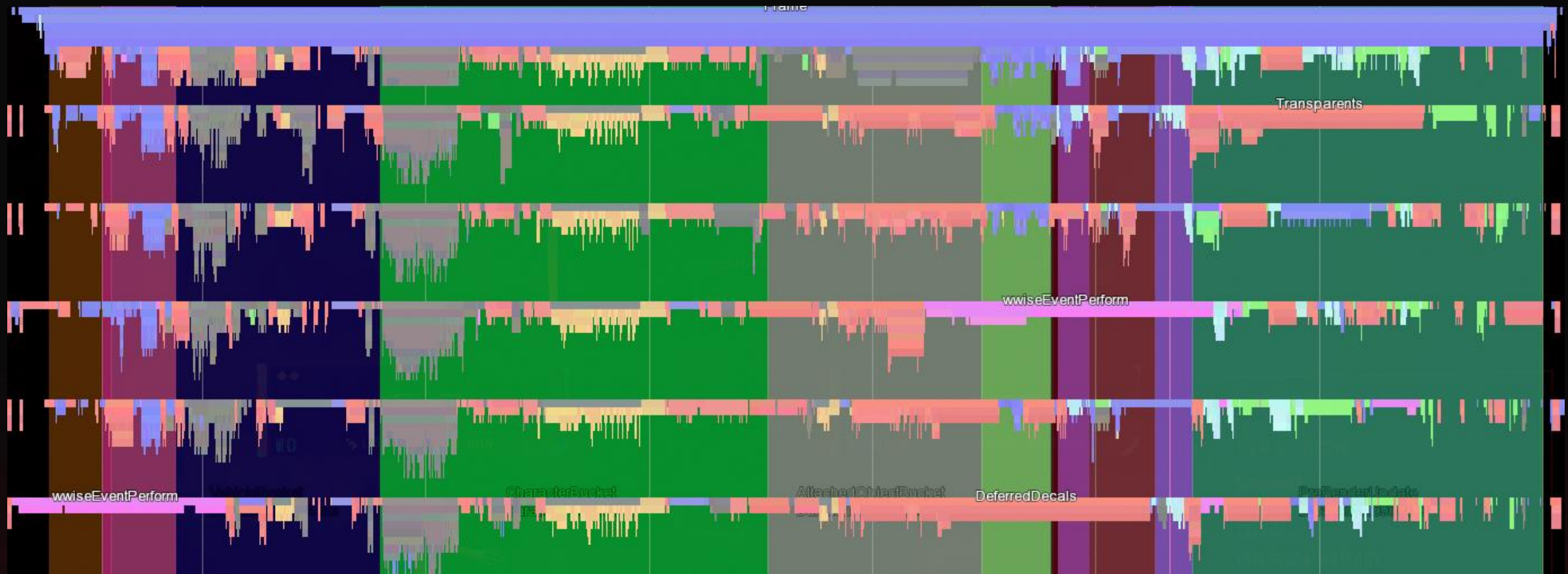
FRAME



BUCKET



Frame Structure – Visualization



Job system

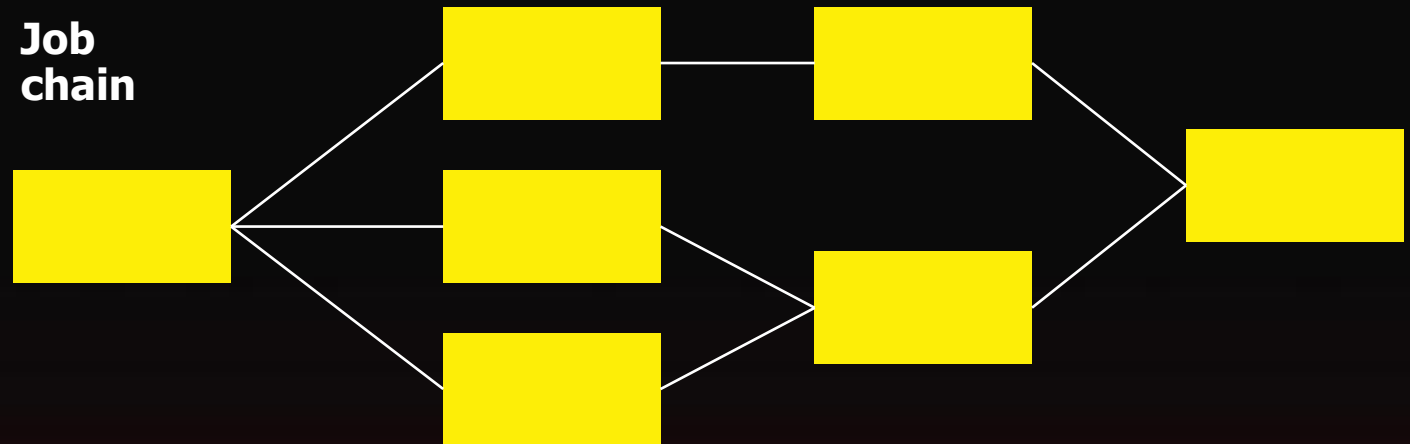
Unshackle Main & Render Thread

Everything should be a job

No more custom thread

Easy to build a job chain

Thousands of jobs per frame



Threads

Main Thread

- Only schedules beginning of frame and waits for completion
- Acts as a regular job worker while waiting

Job Thread / Worker

- Scale well to up to 12

Async IO

Middleware

Telemetry



Systems

World Systems — lifetime limited to world

- Also available for editor preview

Game System — available during the whole game process

Main method to register to frame update

Systems can communicate with each other

- Public interface thread safety needs to be considered



Systems

Any system can register itself to any update group and bucket

By default, systems will be constrained to group / bucket

- However, it is possible to unshackle a system

```
registrar.Register( UpdateTickGroup::PreBuckets, *this, "Audio/Components",
    [this]( const UpdateContext& updateContext, job::Builder& builder ) {
        UpdateComponents( builder, updateContext.m_timeDelta, updateContext.m_tickInfo );
        if( m_isActive ) {
            m_dynamicReverbSystem->Update();
            UpdateEditorHedgehog( engineTime );
        }
    } );

registrar.RegisterInBucket( UpdateBucket::CharacterMask(), UpdateBucketPhase::PrePhysicsTick, *this, "Audio/BatchRaycastResolverCall",
    [this]( UpdateBucket::Enum bucket, const UpdateContext& updateContext, job::Builder& builder ) {
        if( m_isActive ) {
            m_batchRaycastResolver->ResolvePreRaycast();
        }
    } );
```



WORLD STREAMING

World Streaming Systems

WorldStreaming system

- Responsible for streaming "sectors"
- Job Chain Kick start as soon as we have camera position

NodeStreaming system

- Responsible for streaming in & out node/actor/entity
- Job Chain follow up on WorldStreaming Job Graph

Entity system

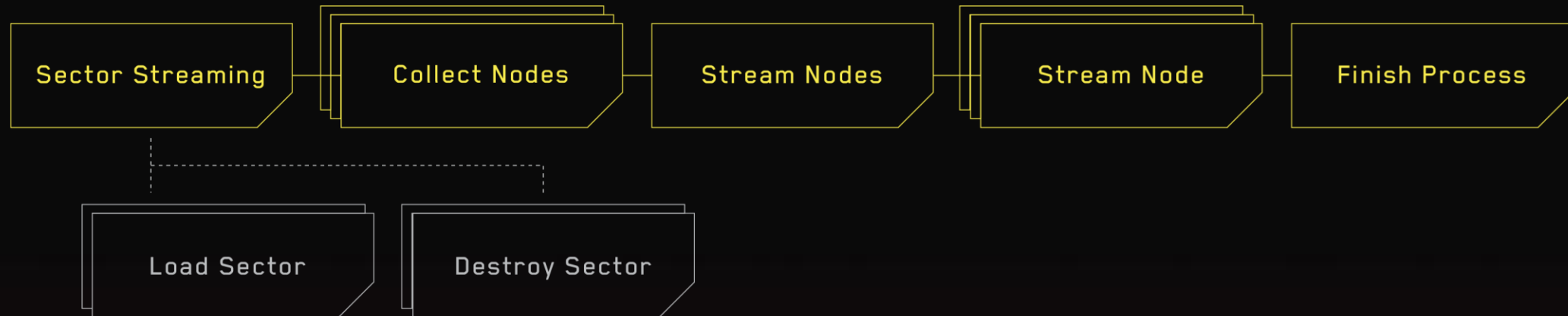
- Attach/Detach entity from world.
- Handle Static and Dynamic spawning



World & Node Streaming – Job Chain

PLAYER
AIM

FRAME
BEGIN

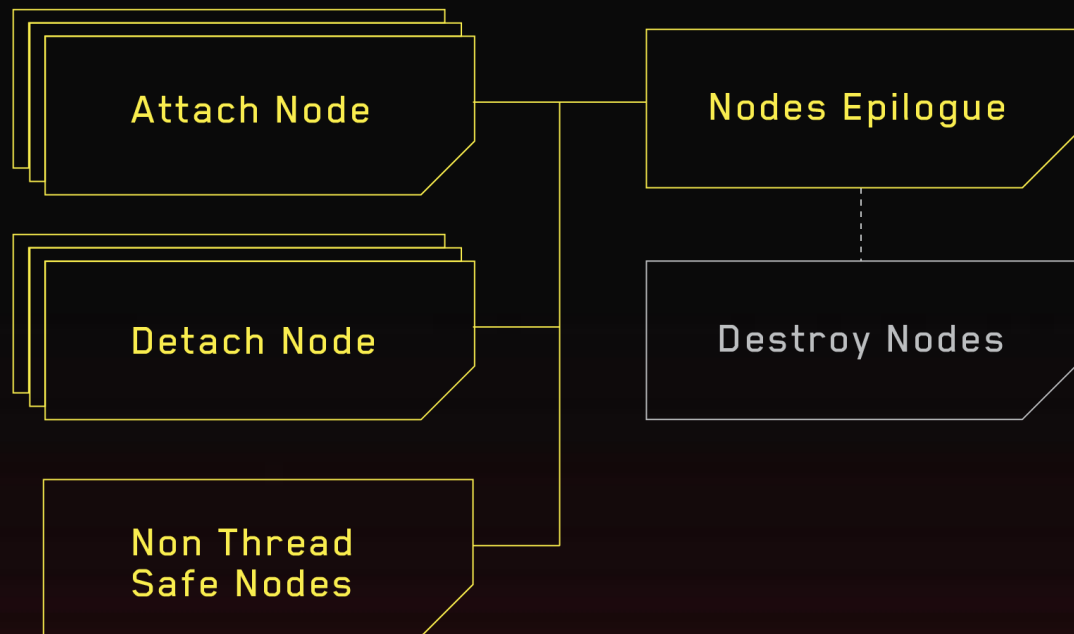


World & Node Streaming – Profiler



Node Management – Job Chain

FRAME
BEGIN



ENTITY
UPDATE
STATE



Entity System – Job Chain

ENTITY
UPDATE
STATE

PREBUCKET

Update Spawn
Request

Transactions

Detach Entity

Attach Entity

Dispose Entities

Entity System – Profiler



What about resource streaming?

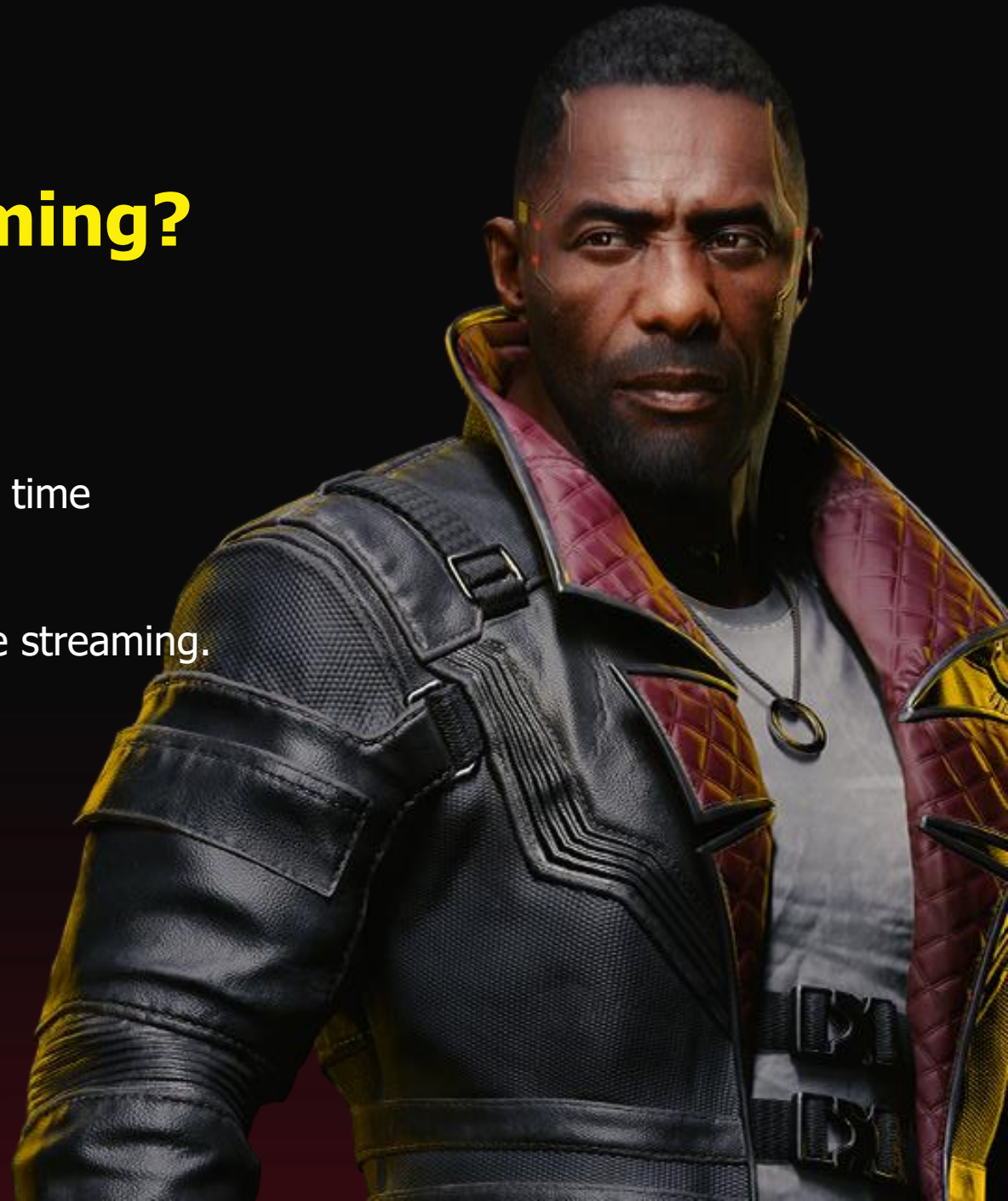
Deserialization & Decompression on Jobs

- At the lowest job priority

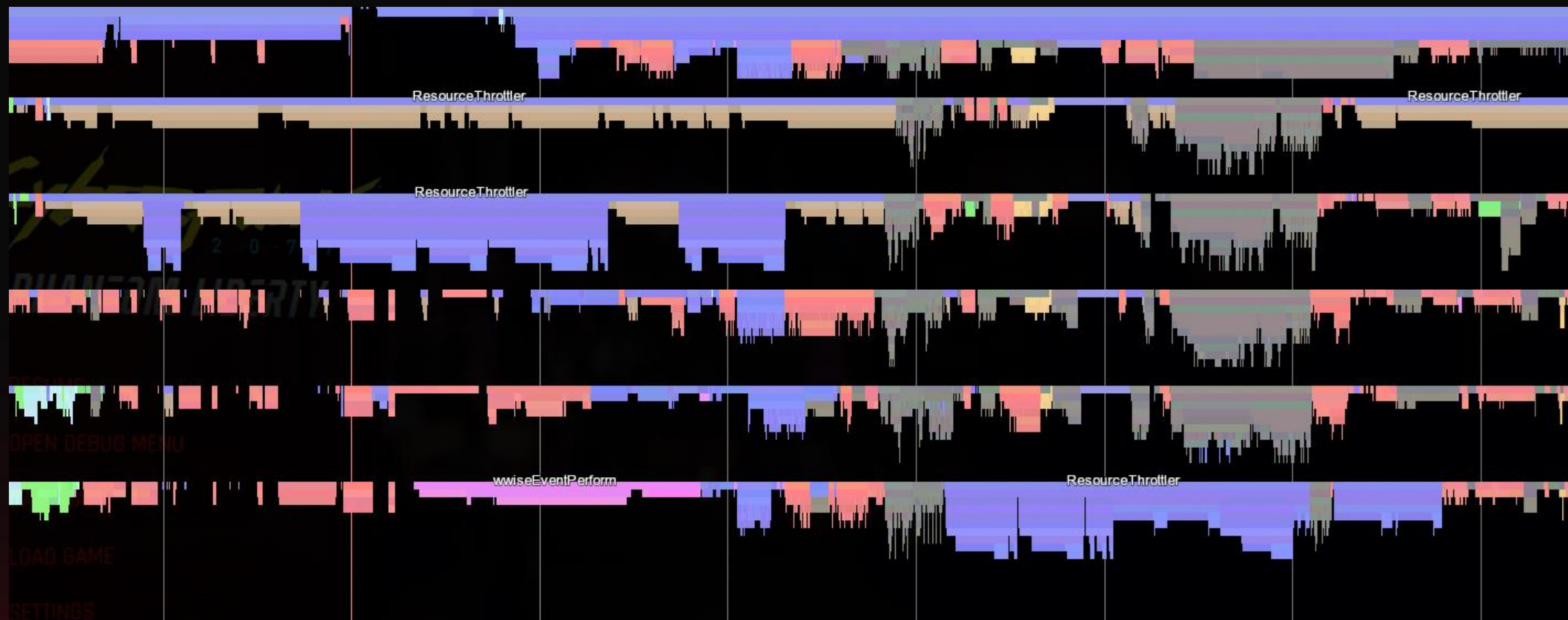
Throttle the amount of resource processed at any given time

All render resources are created asynchronously

No frame dependent job chain can wait on any resource streaming.



Resource Streaming – Job Chain



A cinematic screenshot from the video game Cyberpunk 2077: Phantom Liberty. The scene is set in a dark, industrial environment with a hazy, smoke-filled atmosphere. In the center, a character with a red 'X' over their face and glowing green cybernetic enhancements stands on a set of stairs, holding a yellow handgun. To their left, another character with similar green enhancements and a red 'X' over their face stands with their hands on their hips. In the background, a third character is visible. The foreground is dominated by two large, mechanical, claw-like structures that appear to be part of a larger machine or vehicle. The overall color palette is dark with highlights of green and yellow.

GAMEPLAY

Gameplay Systems

Vehicles System

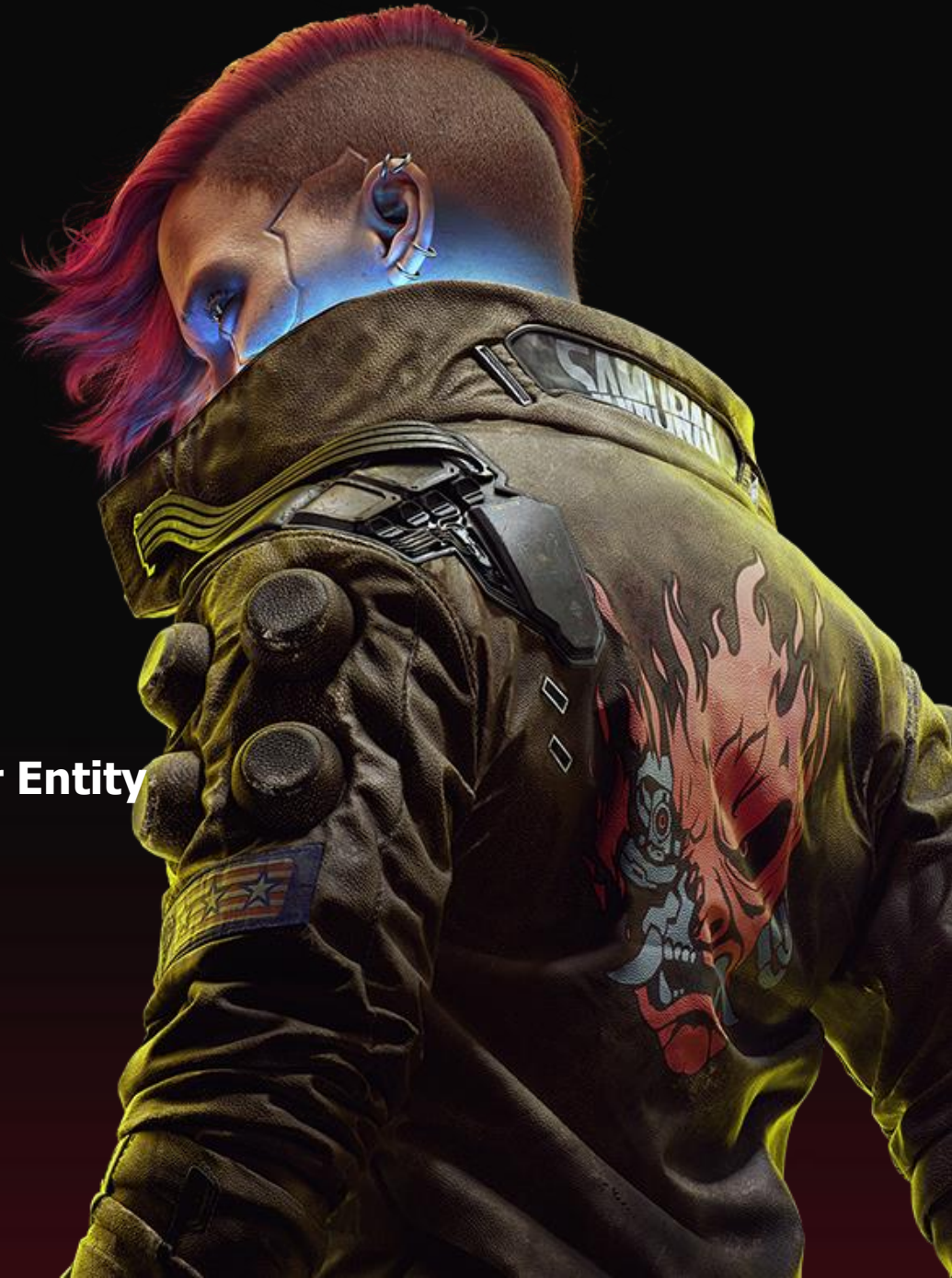
- Update of all vehicle in game
- Vehicle suspension
- NPC Collision

Puppet System

- Complex & simple NPC

Vehicle or NPC cannot communicate directly to another Entity

- Event System was the main communication method
- See last year GDC presentation!



Vehicle System – Job Chain

PRE PHYSICS
TICK

Begin Update

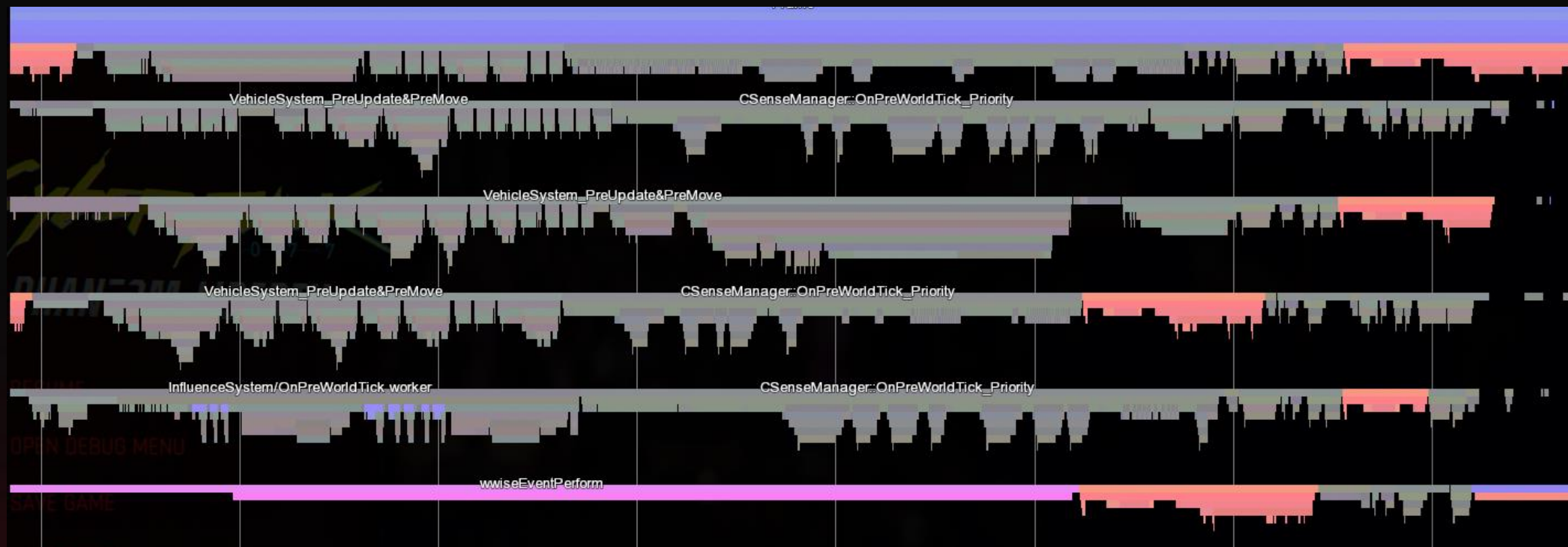
Vehicle Pre Update

Vehicle Fixed Update

Suspension

PRE
PHYSICS
UPDATE
TRANSFORM

Vehicle System – Profiler



Vehicle System – Job Chain

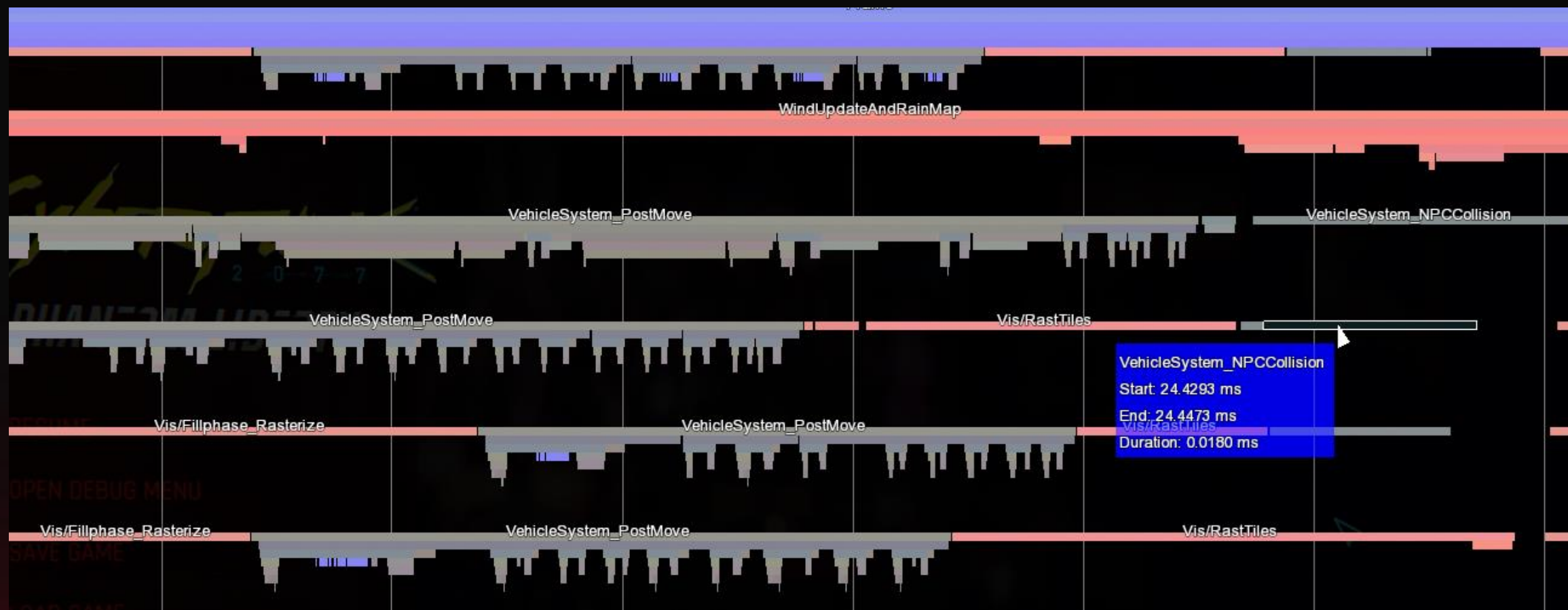
POST PHYSICS
TICK

Vehicle Post Update

NPC Collision

ENTITY
POST TICK

Vehicle System – Profiler



Puppet System – Job Chain

ENTITY
PRE TICK

Complex Puppet Update

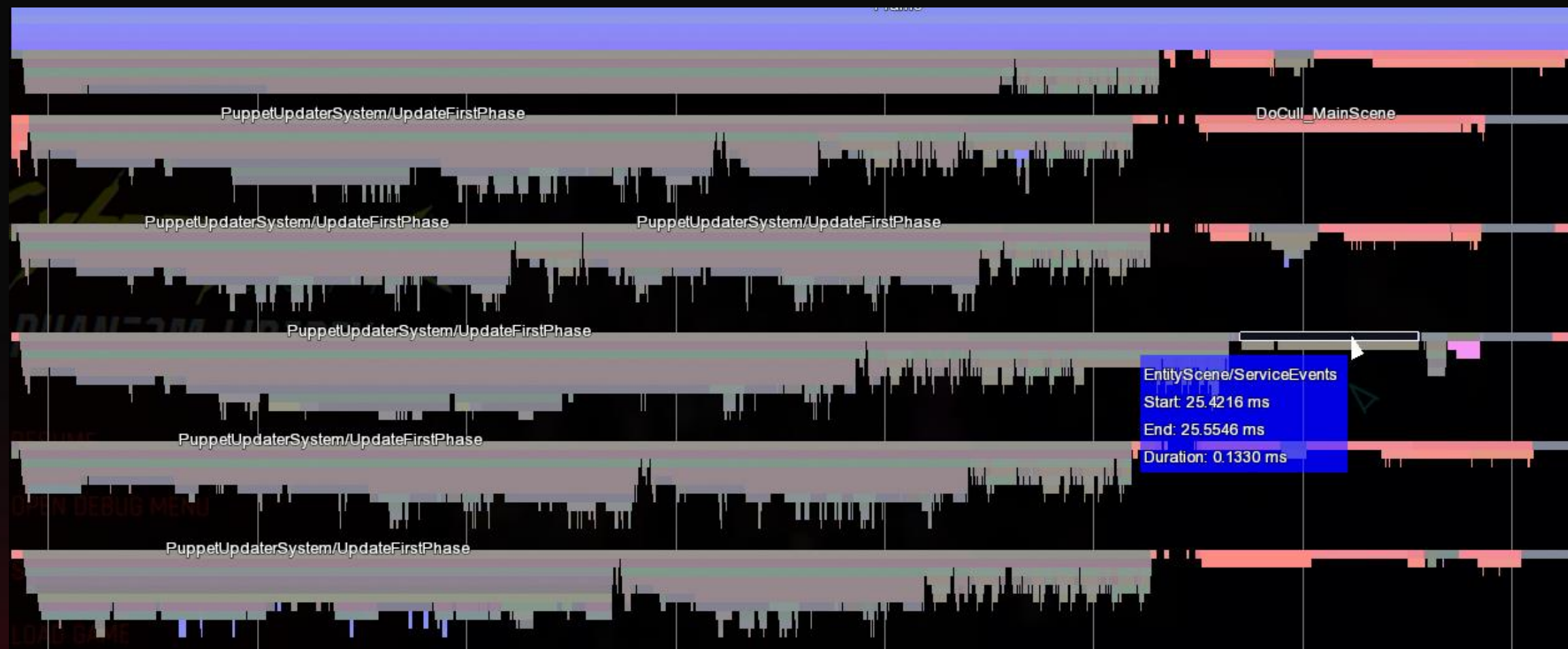
Simple Puppet Update

ENTITY SERVICES
EVENTS

Service Events

PRE PHYSICS
TICK

Puppet System – Profiler





ANIMATION

Animation System

Animation Component register to the Animation System

Parallel update per bucket

Manage Animation instancing for massive standing crowd

No update if occluded. (exception if closer than 5m of player)

Sleep mode for door / vehicles if no movement

Animation Streaming of 40mb budget

- 3k – 4k animations in game



Animation System – Pre Update Job Chain

PRE BUCKETS

Attach/Detach
Animation Components

Attach/Detach
Ragdoll Components

PHYSICS EXECUTE ASYNC QUERIES

Transform Buckets
Update

Update Distance
& Visibility

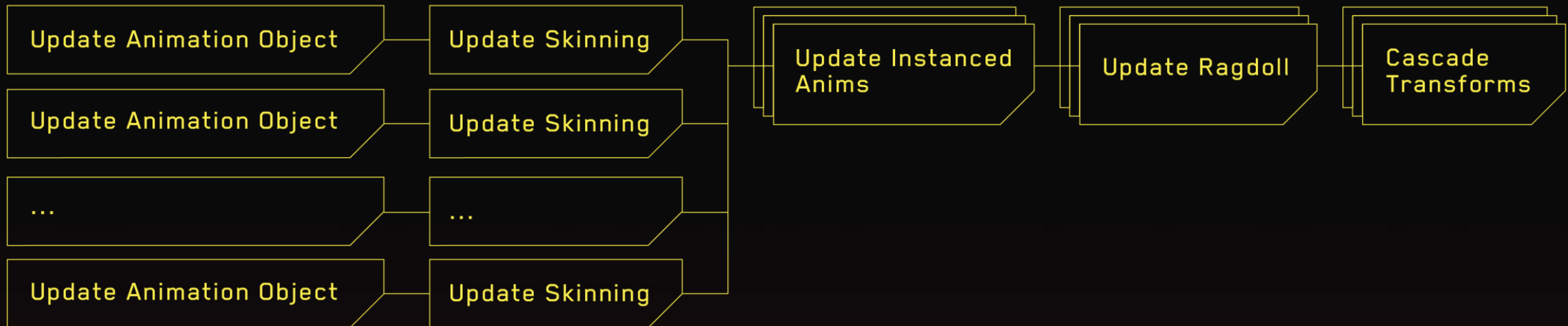
Reserve Skinning
Memory

Update
Controller

UPDATE TRANSFORM POST PHYSICS

Animation System – Update Job Chain

ANIMATION
UPDATE



POST
PHYSICS
TICK

Animation System – Update Job Chain





PHYSICS

Physics System

Manage various Physics Updates

- Collision streamed in and out
- Async Queries
- Simulation

Physic system was built on top of PhysX

PhysX tasks were adapted to be compatible with our job system



Physics System – Job Chain

PRE
BUCKET

Apply WInd

PHYSICS
FLUSH BUFFERED
STATE

Transactions

Step Proxies

PHYSICS
EXECUTE ASYNC
QUERIES

Queries

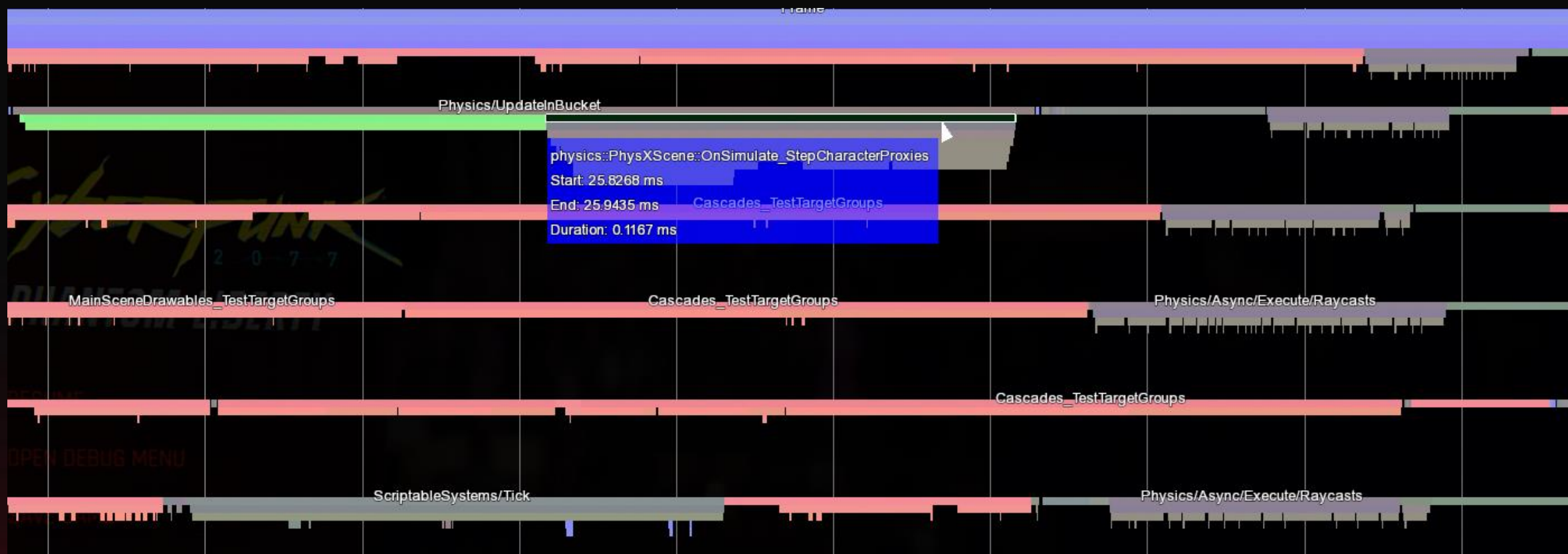
POST PHYSICS
SYNC RESULTS

ANIMATION
UPDATE

Collision
Update

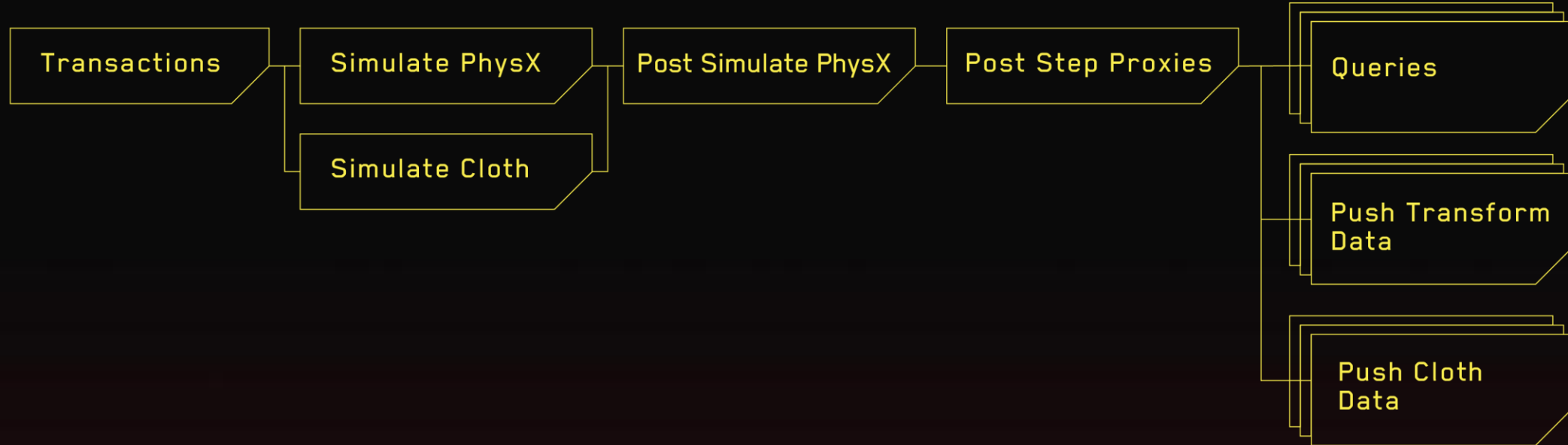
POST PHYSICS
TICK

Physics System – Profiler



Physics Simulation – Job Chain

PRE RENDER
UPDATE



FRAME
BEGIN

Physics Simulation – Profiler





AUDIO

Audio System

Audio Game layer

- Manage interaction between game and audio
- Handle raycast “hedgehog”

Audio System layer

- Double buffer Audio entity
- Transform sound event to “sounds”

Audio Engine & Wwise

- Wwise main processing loop processed by job system
- High Priority!



Audio Game Layer – Job Chain

PRE
BUCKET

Update Audio
Components

PRE PHYSICS
TICK

Perform Async
Raycast

POST PHYSICS
SYNC RESULT

Process Raycast
Results

POST
PHYSICS
UPDATE
TRANSFORM

Audio System Layer – Job Chain

PRE
RENDER
UPDATE

Update Audio
Entity

Update
Sub Systems

Accoustics
Systems Update

FRAME
BEGIN

The background of the slide is a dark, atmospheric image of a character from the video game Cyberpunk 2077: Phantom Liberty. The character is a man with a serious expression, wearing a dark jacket and a tactical vest. He has a cybernetic arm on his left side. The scene is dimly lit with some blue and purple ambient light in the background.

GRAPHICS

Graphics – Render Graph

Breaking up the Monolithic Render Thread

- Fully parallelized, using job system

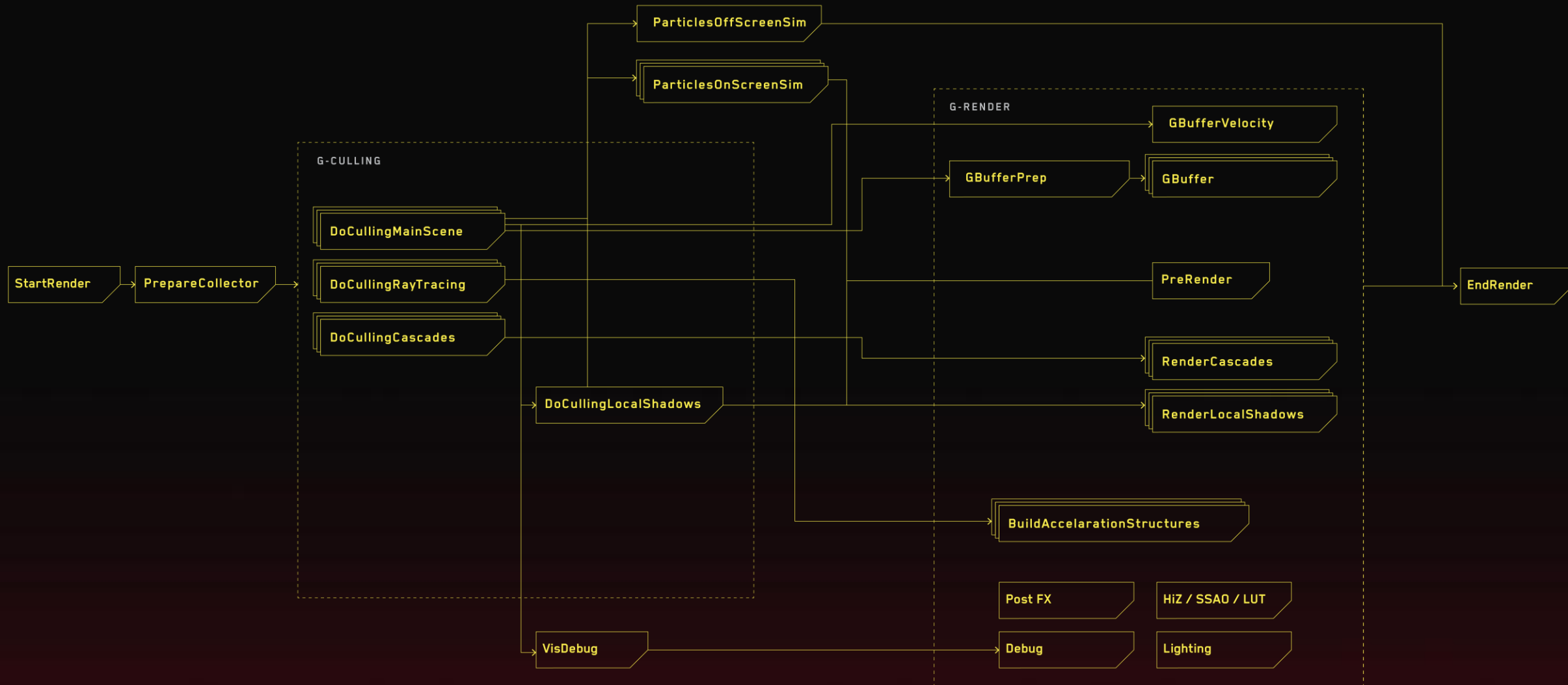
Multiple graphs for different use cases

Runs across threads at lower priority than game jobs

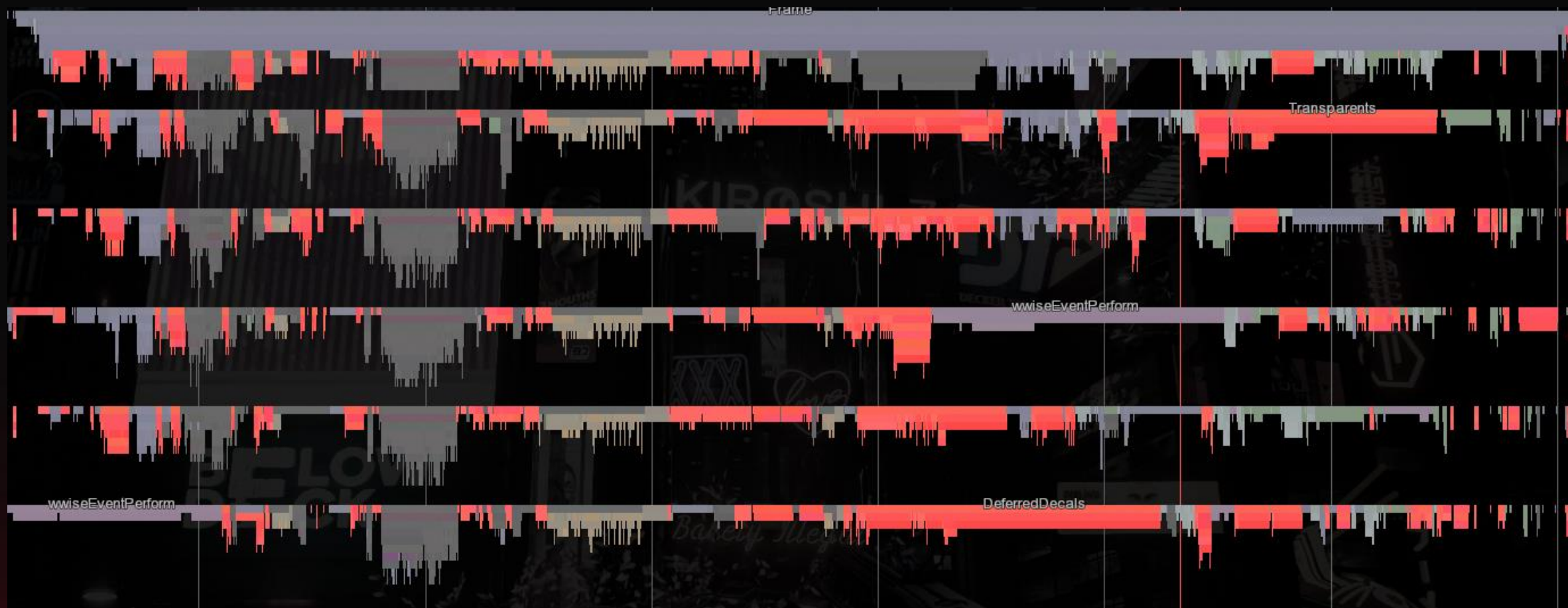
Fill nicely the “gap” in frame



Graphics – Render Graph



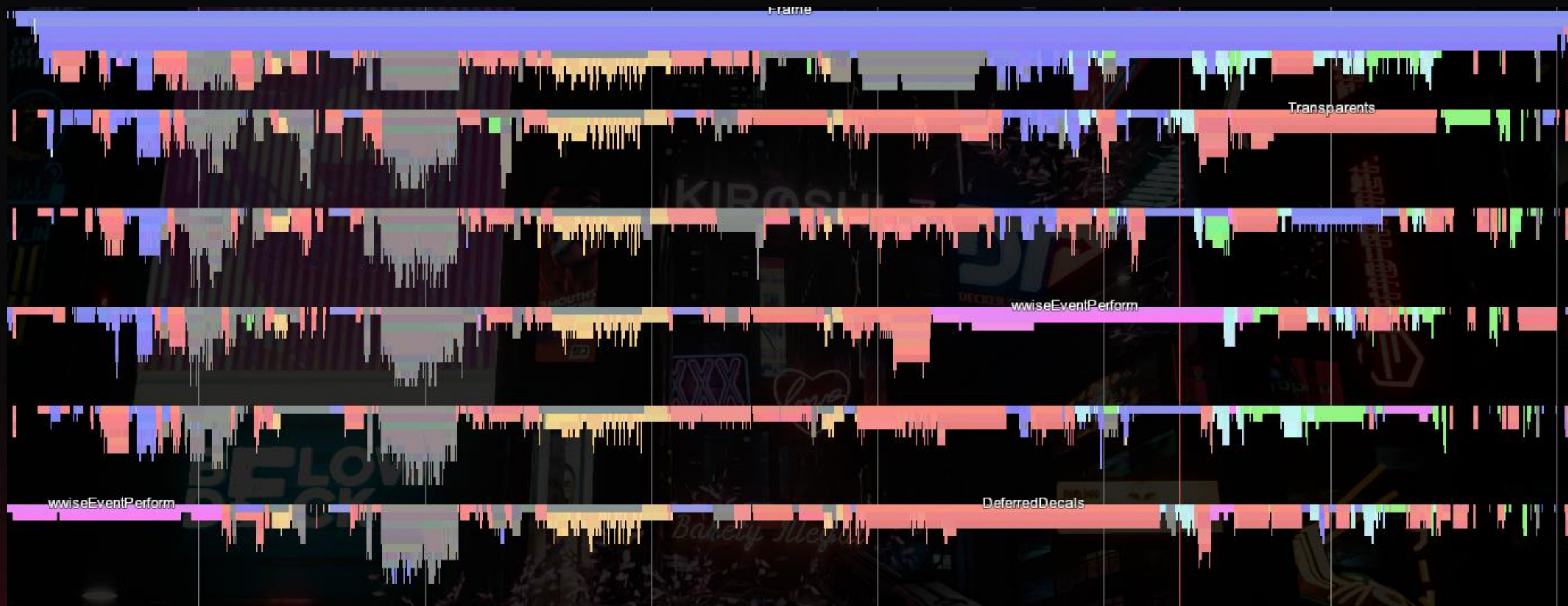
Graphics – Profiler



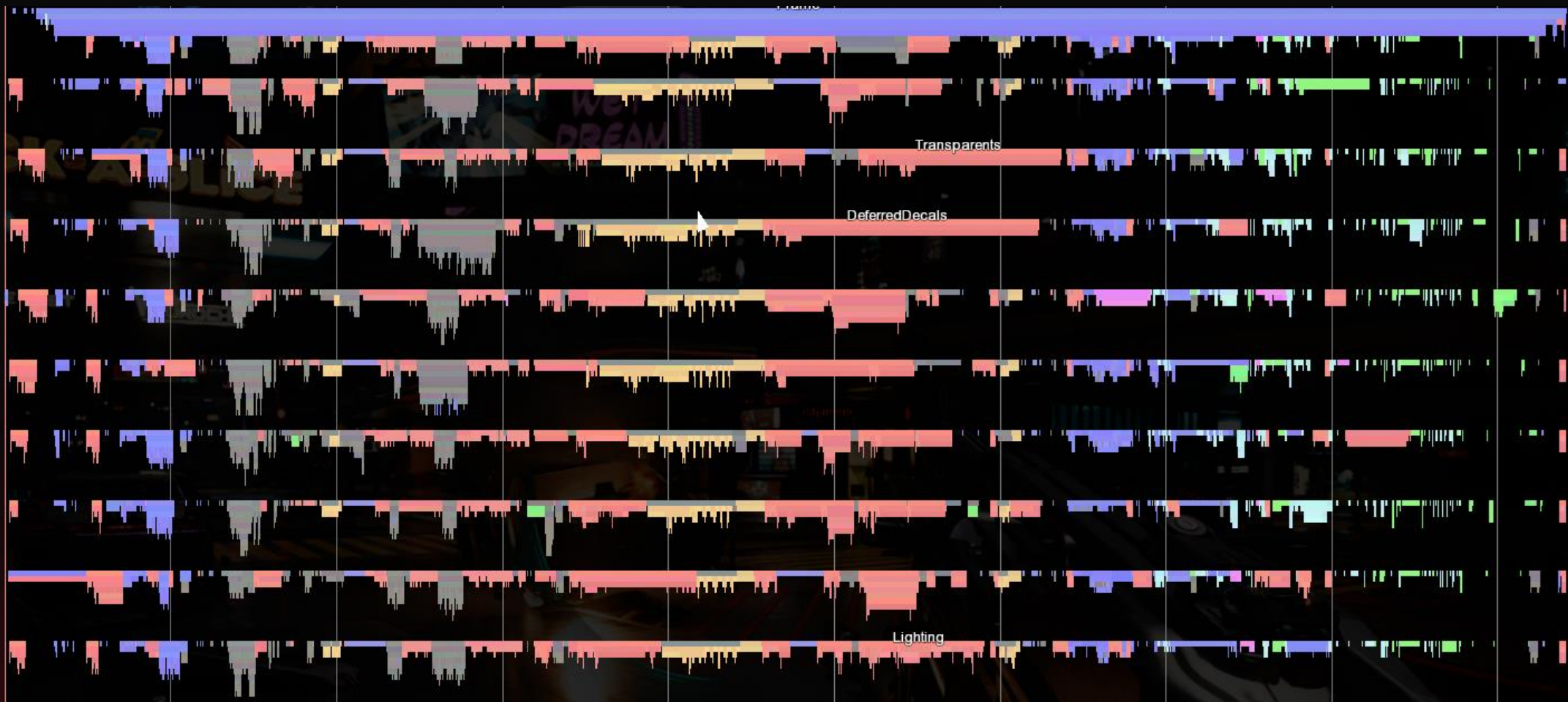
A character from Cyberpunk 2077: Phantom Liberty is shown sitting in the driver's seat of a vehicle. The character is wearing a light-colored, high-collared jacket with a dark belt and matching pants. They are also wearing dark sunglasses and have a small earring. The vehicle's interior is visible, including the red and white patterned seat and a control panel with many buttons. The background is a blurred, dark, industrial or urban setting. The text "PERFORMANCE & SCALABILITY" is overlaid in the center in a bold, yellow, sans-serif font, enclosed in a thin yellow rectangular border.

PERFORMANCE & SCALABILITY

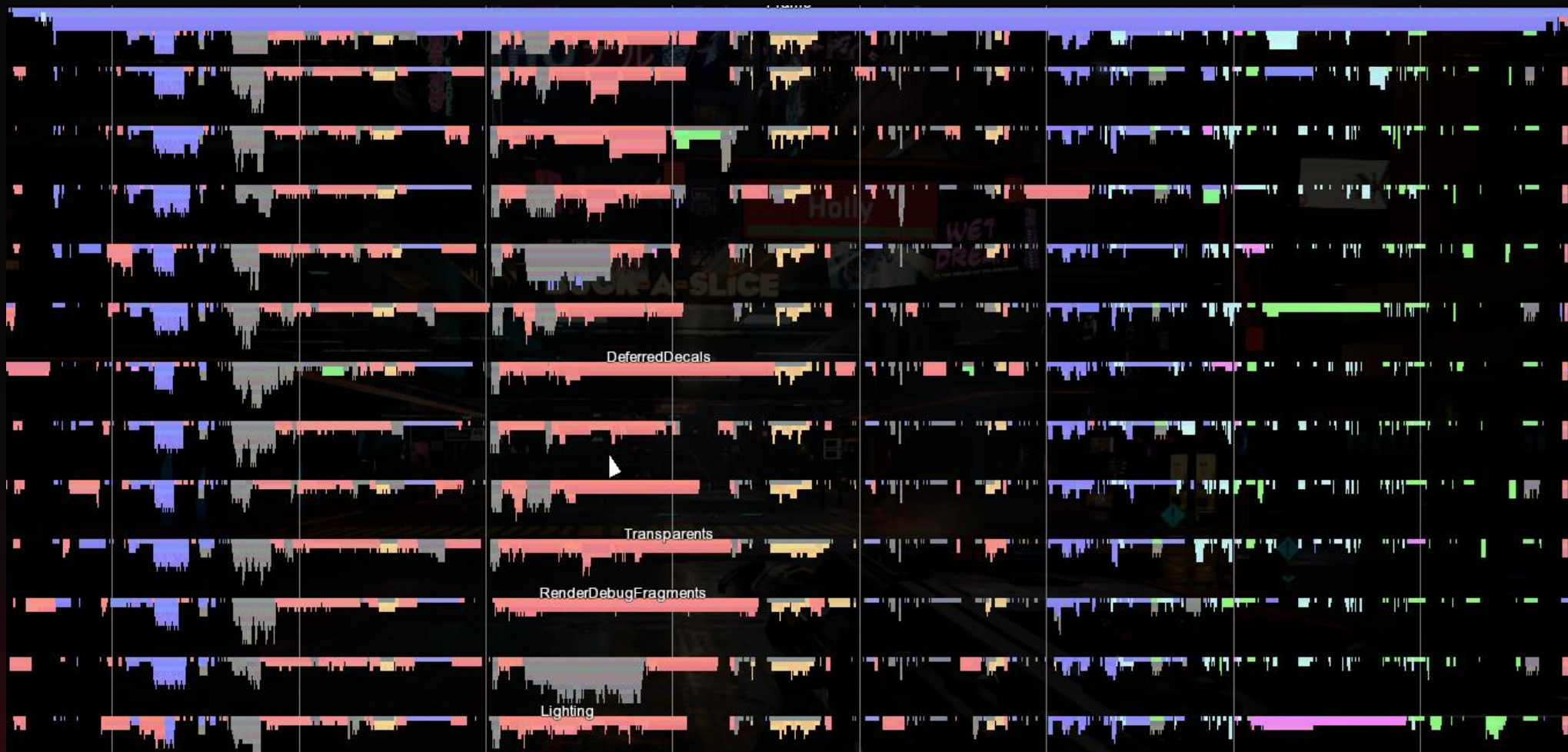
Scalability – 5 Job Worker



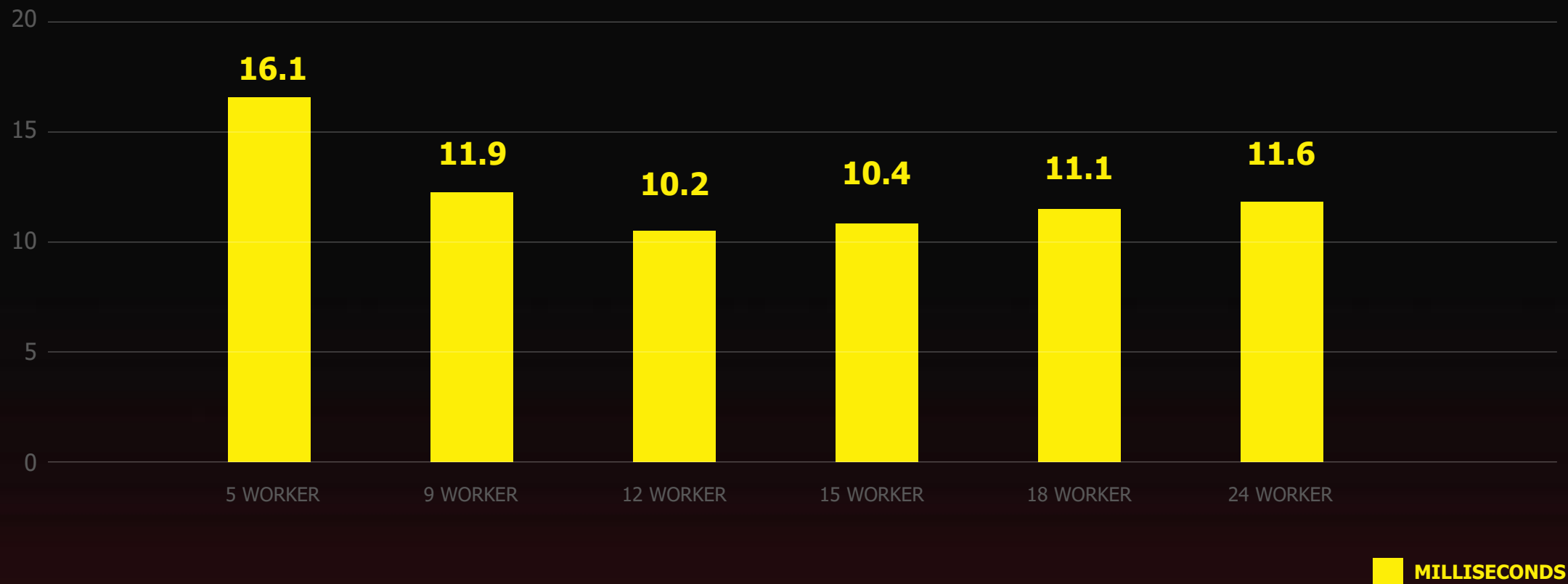
Scalability – 9 Job Worker



Scalability – 12 Job Worker



CPU Frame Time



A cinematic scene from Cyberpunk 2077: Phantom Liberty. V, a cybernetically enhanced character, stands on the left in a dark, industrial setting. Dog, a large, cybernetic dog, is crouching on the right. The background features a control room with a person at a console and various industrial elements. The scene is dimly lit with blue and green ambient light.

CONCLUSION

Frame Structure – First Iteration

What Went Wrong?

Frame structure & Budgets were defined late
Designing system to be scalable is hard
Refactoring to be scalable in production was painful.

Frame Structure – Second Iteration

What Went Right?

Simple frame structure that was easy to understand

Everything was multithreaded

- Including gameplay & rendering

Scale very well on CPU across all platforms



THANK YOU CHOOMS

**MATHIEU GIRARD**

Creating the Blackwall VFX for
'Cyberpunk 2077: Phantom Liberty'

Mon, March 18, 13:20

**KAJETAN KASPROWICZ**

Cinematic and Interactive
– Two Sides of the Same Coin

Mon, March 18, 15:00

**AMELIA KOŁAT
SEBASTIAN SIEJKA**

Blurring the Border Between Reality
and Cyberpunk 2077 using ARG

Mon, March 18, 15:50

**ANIA BULAVINA**

Urban Planning in Games

Tue, March 19, 9:30

**KACPER NIEPOKÓLCZYCKI
PAWEŁ MIELNICZUK**

Deciphering Dogtown: How We Built
the Last District of Night City

Tue, March 19, 16:10

**SARAH GRUEMMER**

Where Story and Tech Meet:
Cyberpunk's Quest Editor in Action

Wed, March 20, 14:00

**COLIN WALDER
GIUSEPPE MARANO**

Mixing, Music, and Mods:
Cyberpunk 2077 Audio Technical Pipelines

Wed, March 20, 15:30

**KAROLINA NIEWĘGŁOWSKA**

A Ticket to Redemption: How CD PROJEKT RED
Utilized Technical Support Data and Insights
to Improve Cyberpunk 2077

Thu, March 21, 10:00

**DAVID BLOCK**

Scaling Night City on the CPU

Thu, March 21, 11:30

**ALEX RADKEVICH**

The Key Lessons of Cyberpunk 2077:
Phantom Liberty Localization

Thu, March 21, 11:30

**PAWEŁ SASKO**

Designing AAA Expansions
for Cyberpunk 2077 and The Witcher 3

Thu, March 21, 17:30

**CHARLES TREMBLAY**

The Anatomy of a Frame:
Cyberpunk 2077 Phantom Liberty

Fri, March 22, 10:00

**MAGDA ZYCH**

Make a Choice and See What Happens:
CD PROJEKT RED's Dialogue System

Fri, March 22, 15:00



WANT TO WORK WITH US?

Our Boston team is growing in strength, with numerous industry veterans joining the team.

Check out our available positions to find the right fit for your expertise!

Scan the QR code and head over to our page.

