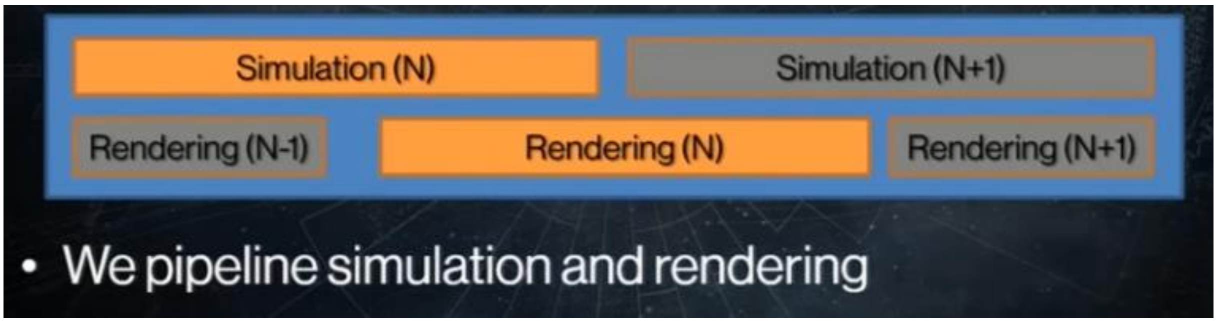
Multithreading in Games

Multithreading the Entire Destiny Engine

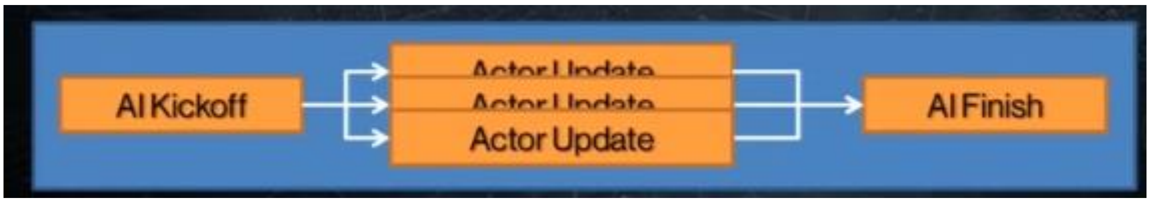
Destiny Rendering

- Simulation und Rendering können getrennt werden

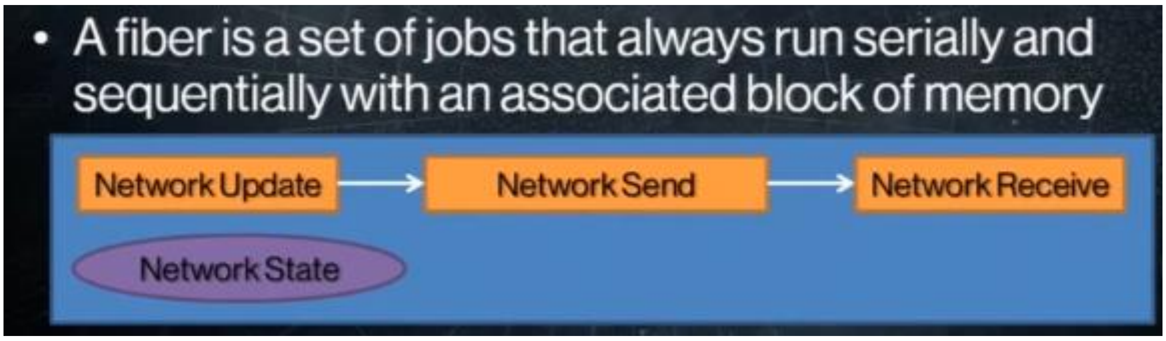


- We pipeline simulation and rendering

- Job System – Ideal duration 500us – 2000us
- Dependencies between Jobs



- Misused 'Fiber'



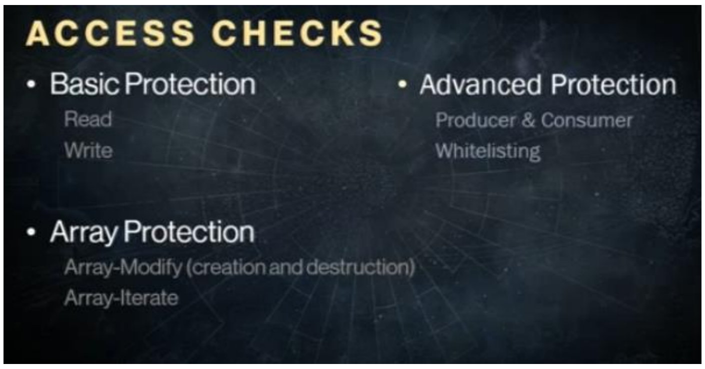
- A fiber is a set of jobs that always run serially and sequentially with an associated block of memory

- Small set of Fibers (Simulation, some Rendering, Networking)
- Eigener Thread für TimeControl serialisiert alles außer async jobs
-

Thread safety

- Policies for access of Recources – associated with a Job
- Look in the policy to see if access is allowed
- Define which policies are allowed at the same time
- -> Determine which Jobs can run at the same time

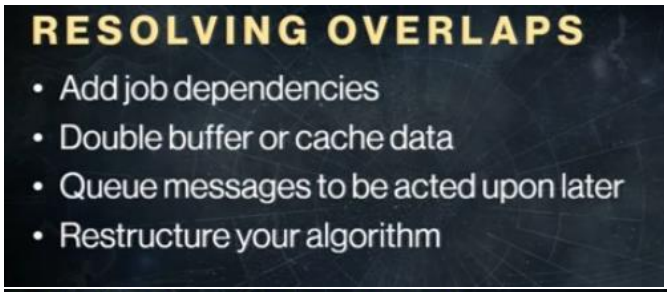
- Handles for data access



ACCESS CHECKS

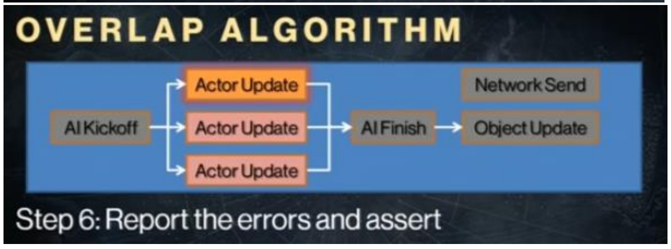
- Basic Protection
  - Read
  - Write
- Advanced Protection
  - Producer & Consumer
  - Whitelisting
- Array Protection
  - Array-Modify (creation and destruction)
  - Array-Iterate

Resolving Overlaps

- 

RESOLVING OVERLAPS

- Add job dependencies
- Double buffer or cache data
- Queue messages to be acted upon later
- Restructure your algorithm

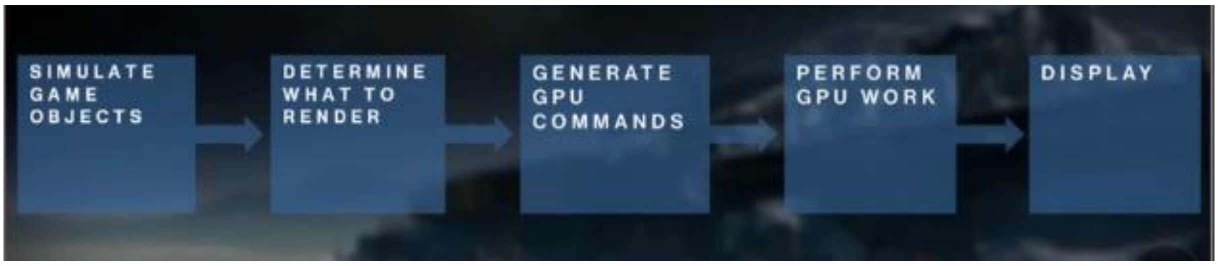
- 

OVERLAP ALGORITHM

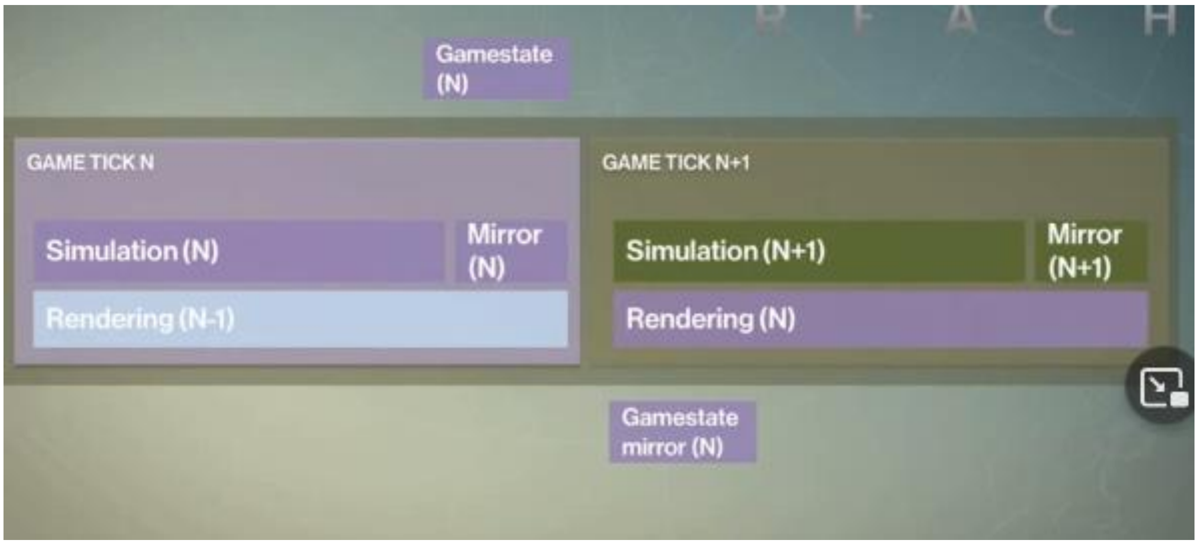
Step 6: Report the errors and assert

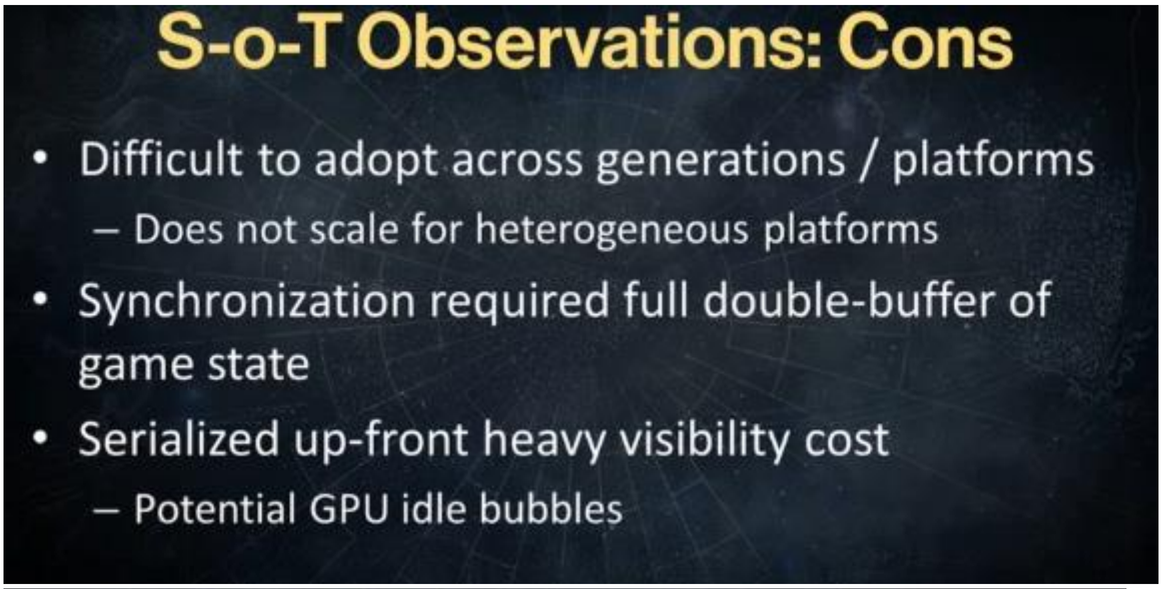
- Enable Overlaps first

Genereller Ablauf des Game Loops



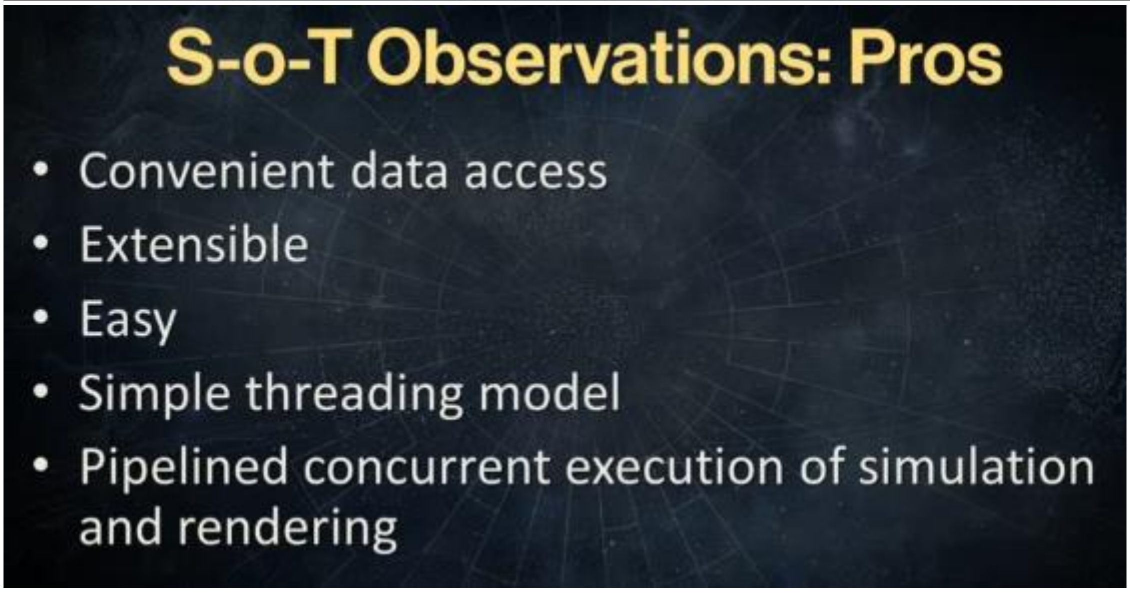
Gameloop bei festen Threads pro System (Simulation, Render, Audio, Job Kernel, Misc)





S-o-T Observations: Cons

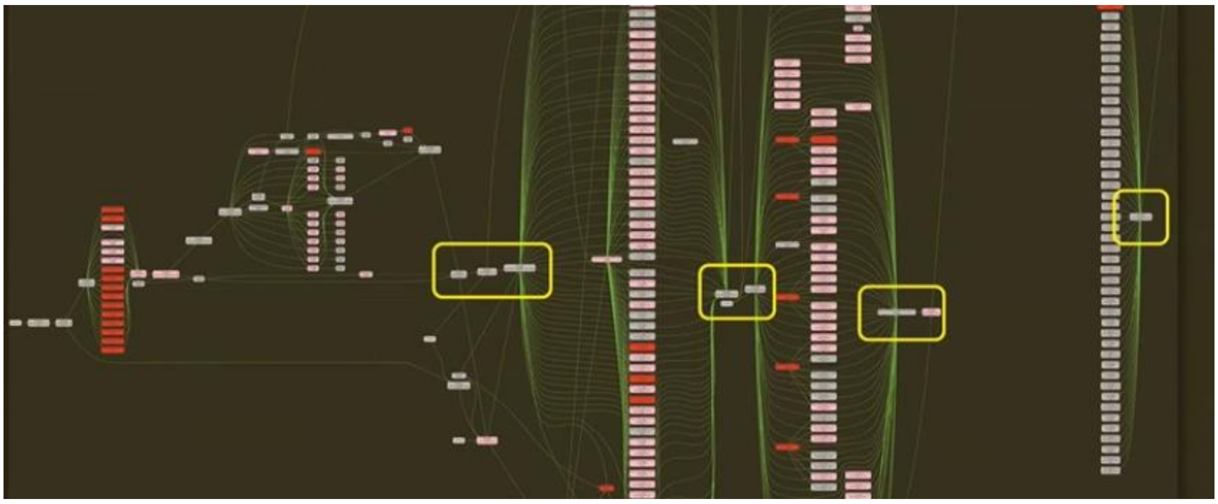
- Difficult to adopt across generations / platforms
  - Does not scale for heterogeneous platforms
- Synchronization required full double-buffer of game state
- Serialized up-front heavy visibility cost
  - Potential GPU idle bubbles



S-o-T Observations: Pros

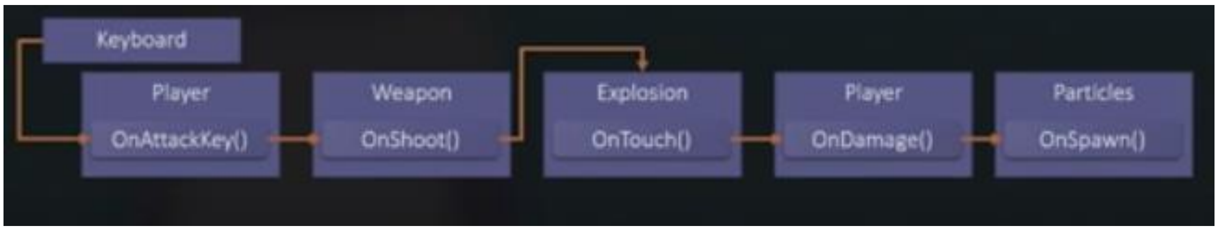
- Convenient data access
- Extensible
- Easy
- Simple threading model
- Pipelined concurrent execution of simulation and rendering

Definierte Synchronisierungspunkte



Definition von Immutable states

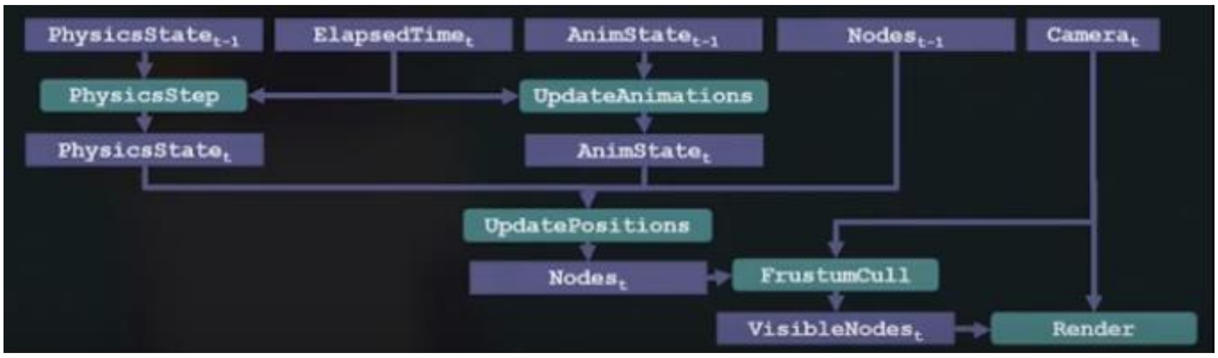
Vermeidung von Spaghetti flow



Viele Systeme werden angefasst -> Definition von “Todo” Lists für Batching

- ➔ Definition einer Reihenfolge von Systemen/Jobs

Definierte Synchronisierungspunkte


PhysicsState$_{t-1}$
ElapsedTime$_t$
AnimState$_{t-1}$
Nodes$_{t-1}$
Camera$_t$
PhysicsStep
UpdateAnimations
PhysicsState$_t$
AnimState$_t$
UpdatePositions
Nodes$_t$
FrustumCull
VisibleNodes$_t$
Render