AWS re:Invent

CMP328-R

How Uber builds efficient & scalable autonomous vehicle simulations on AWS Batch

Jo Adegbola
Sr. Manager, Software Development
AWS

Matt Ranney
Senior Staff Engineer
Uber

Steve Kendrex
Sr. Product Manager
AWS

AWS re:Invent

aws

Broadest and deepest platform choice

Categories	Capabilities	Options
General purpose	Choice of processor (AWS, Intel, AMD)	Amazon Elastic Block Store (Amazon EBS)
Burstable	Fast processors (up to 4.0 GHz)	Amazon Elastic Inference
Compute intensive	High memory footprint (up to 12 TiB)	
Memory intensive	Instance storage (HDD and NVMe)	
Storage (high I/O)	Accelerated computing (GPUs and FPGA)	
Dense storage	Networking (up to 100 Gbps)	
GPU compute	Bare Metal	
Graphics intensive	Size (Nano to 32xlarge)	

270+

instance types

for virtually every workload and business need

An introduction to AWS Batch

AWS re:Invent

aws

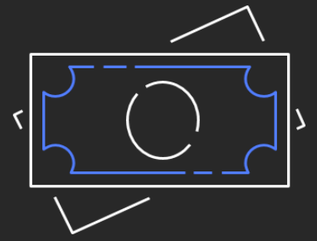
Introducing AWS Batch

Fully managed

No software to install or servers to manage. AWS Batch provisions, manages, and scales your infrastructure.

Integrated with AWS services

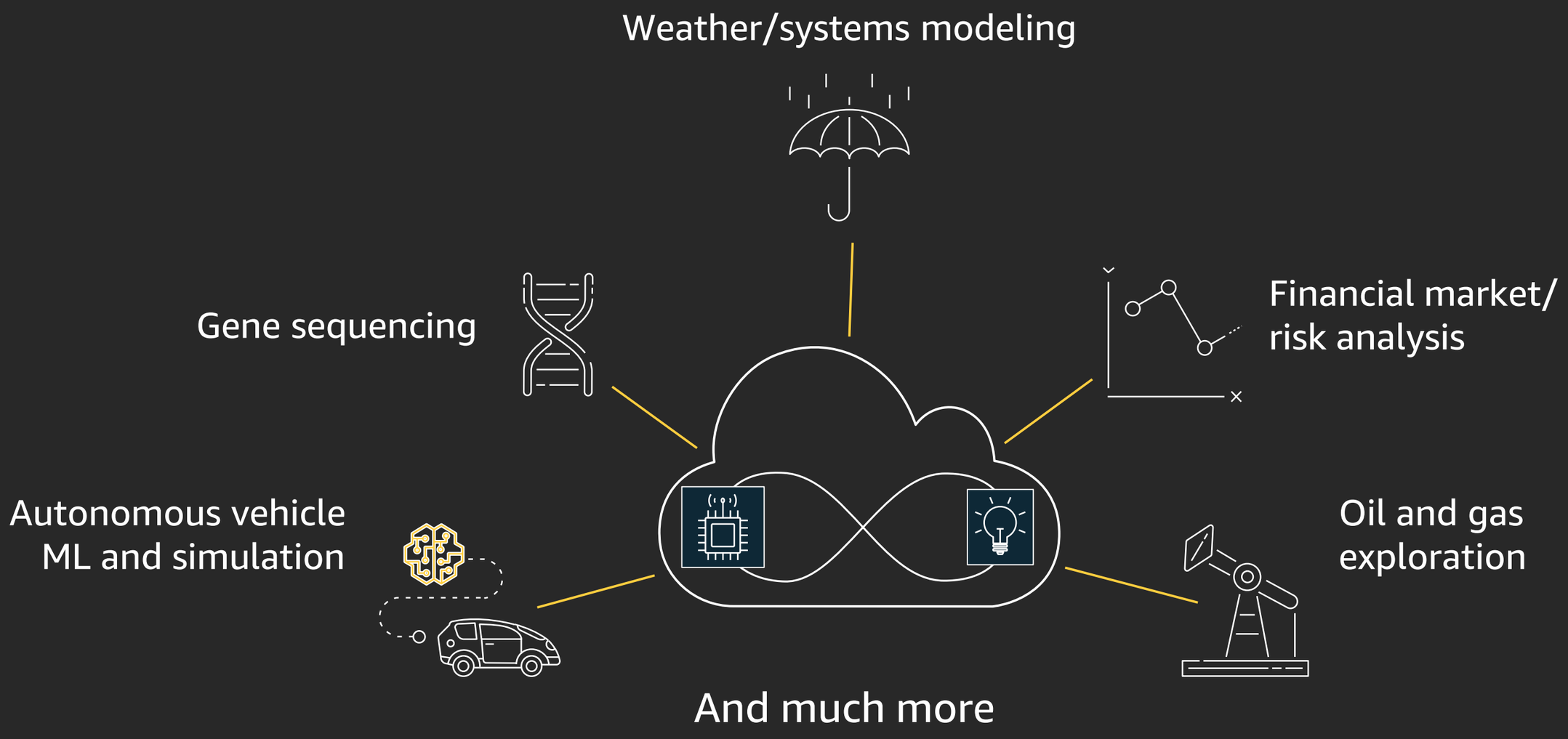
Natively integrated with the AWS platform, AWS Batch jobs can easily and securely interact with services such as Amazon Simple Storage Service (Amazon S3), Amazon DynamoDB, and Amazon Rekognition



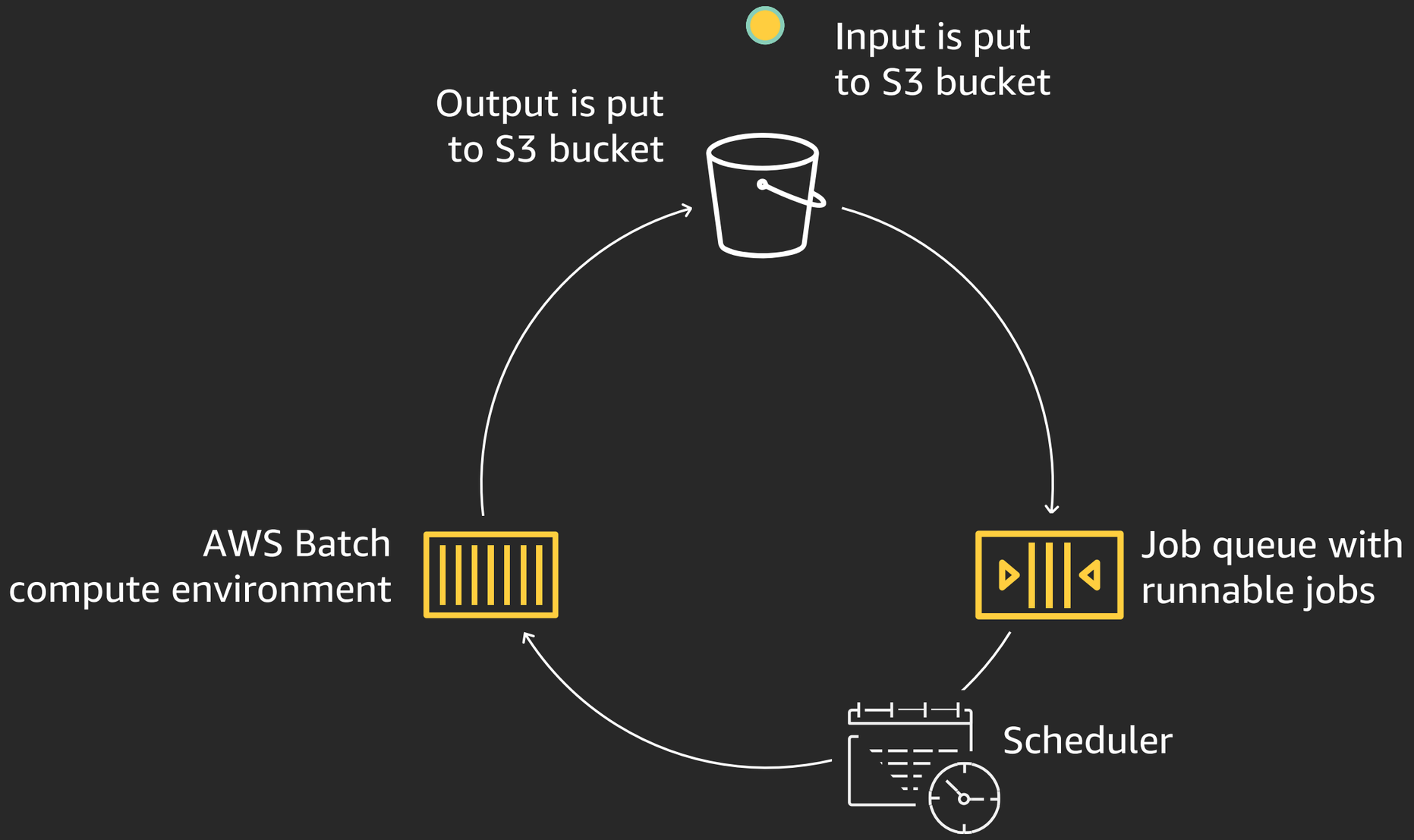
Optimized resource provisioning

AWS Batch automatically provisions compute resources tailored to the needs of your jobs using Amazon Elastic Cloud Compute (Amazon EC2) and Amazon EC2 Spot

Who uses AWS Batch



Typical AWS Batch job architecture



Job definition

Application
image + config

IAM role

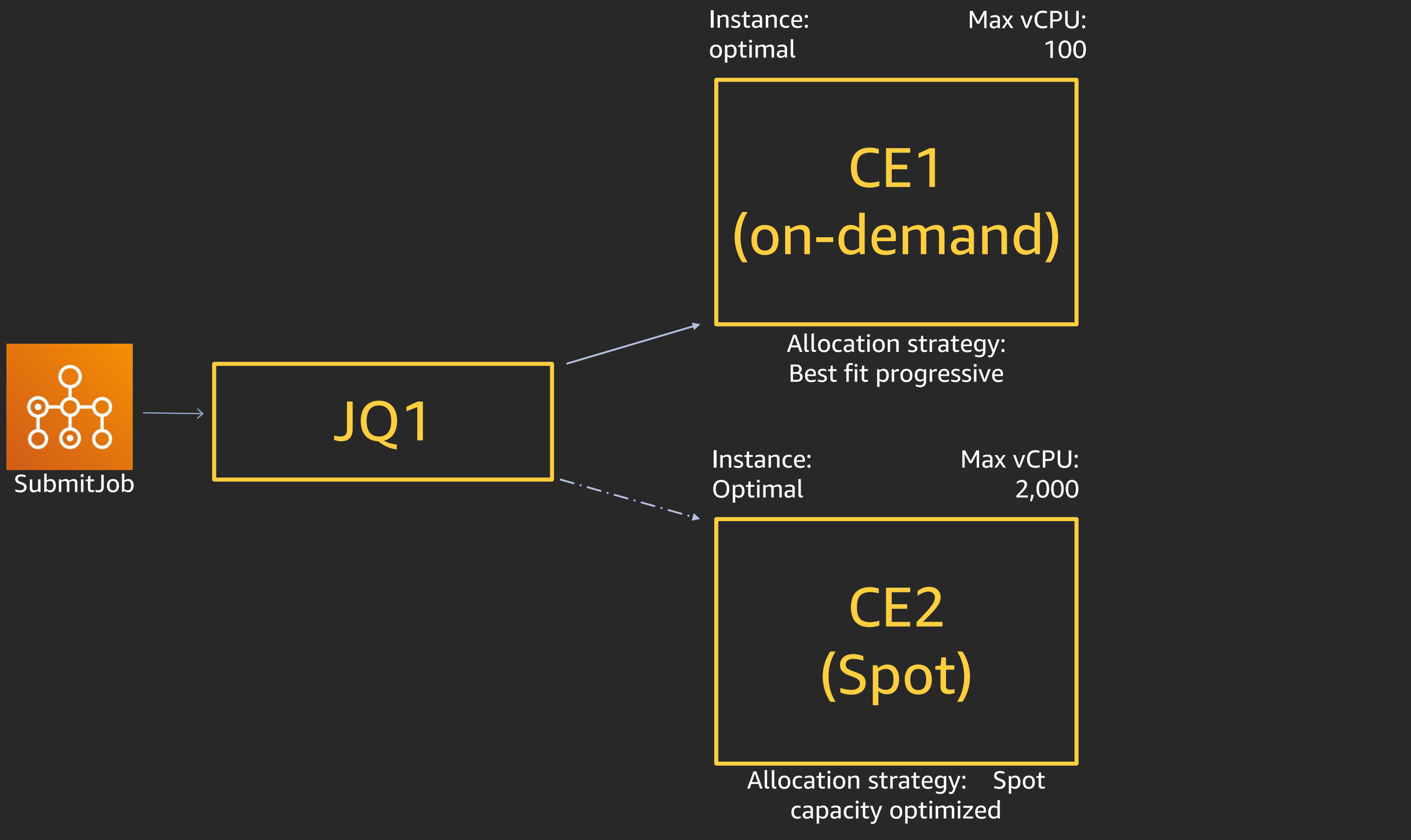
New: Allocation strategies for AWS Batch

Make capacity/throughput/cost tradeoffs

Spot Capacity Optimized—Allow AWS to predict the deepest Spot capacity pools, and launch instances accordingly. Available in Spot only.

Best Fit—Same behavior as previously: CEs that are created through SLI/SDK will default to this (to preserve backwards compatibility). Spot or On-demand CE's supported. Not recommended for most use cases.

Best Fit Progressive—Same as Best Fit, but when we reach a capacity error (ICE, Reclaim, EC2 Limit), AWS Batch will progressively sort through the list and pick the next best fit instance type. Recommended for OD CEs or in Spot CEs with a specific use case.

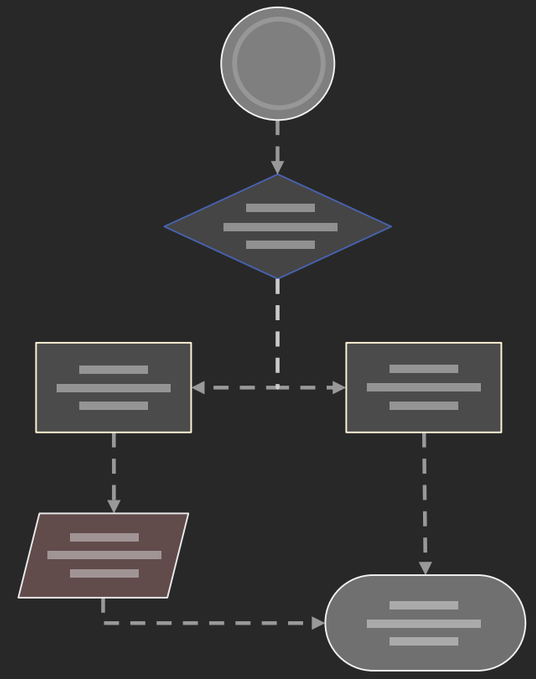

Instance:
optimal
Max vCPU:
100
CE1
(on-demand)
Allocation strategy:
Best fit progressive
SubmitJob
JQ1
Instance:
Optimal
Max vCPU:
2,000
CE2
(Spot)
Allocation strategy: Spot
capacity optimized

How does this fit together for ML and simulations

Workflows, pipelines, and job dependencies

Jobs can express a dependency on the successful completion of other jobs or specific elements of an array job

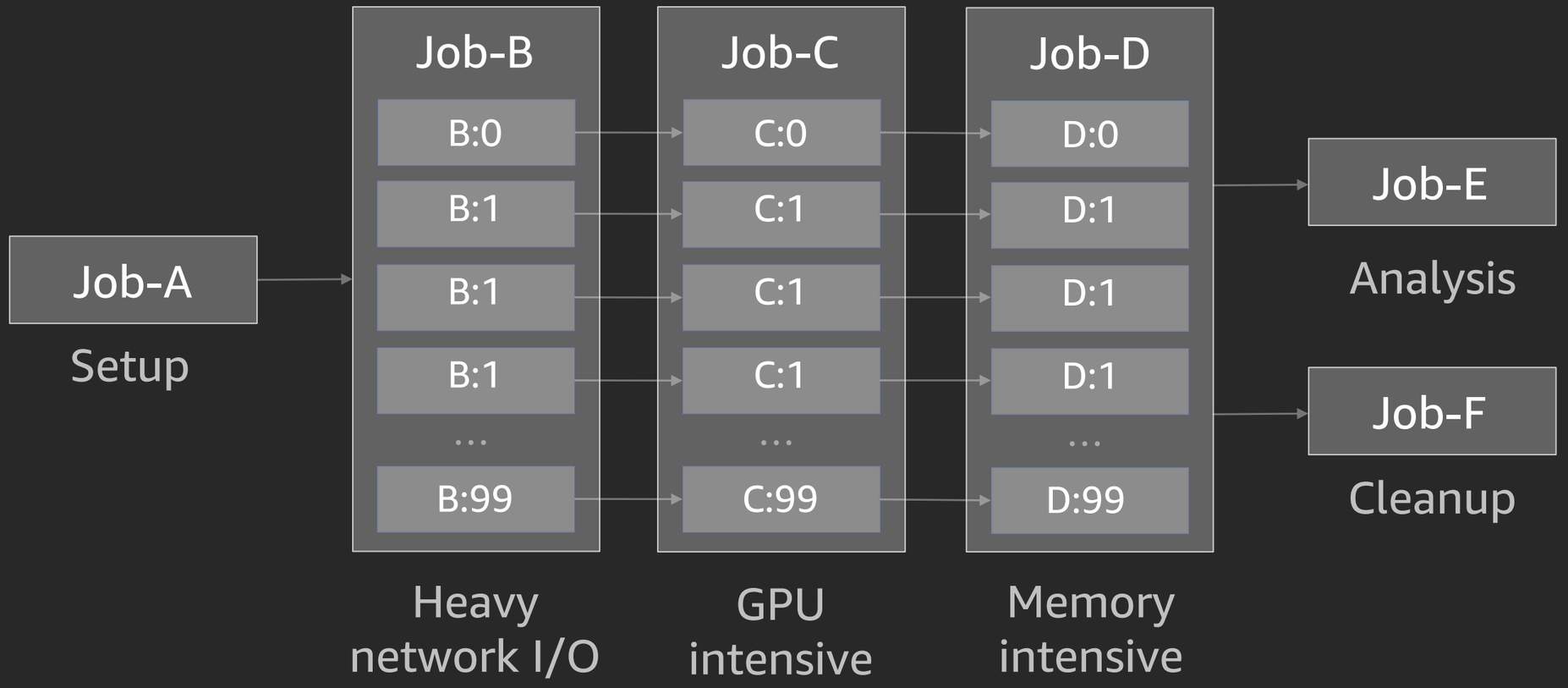
Use your preferred workflow engine and language to submit jobs. Flow-based systems simply submit jobs serially, while DAG-based systems submit many jobs at once, identifying inter-job dependencies.



```
aws batch submit-job --depends-on 606b3ad1-aa31-48d8-92ec-f154bfc8215f
```

Model example

C is dependent on A,
C N_TO_N dependency on B, same for D an C,
E and F depend on D



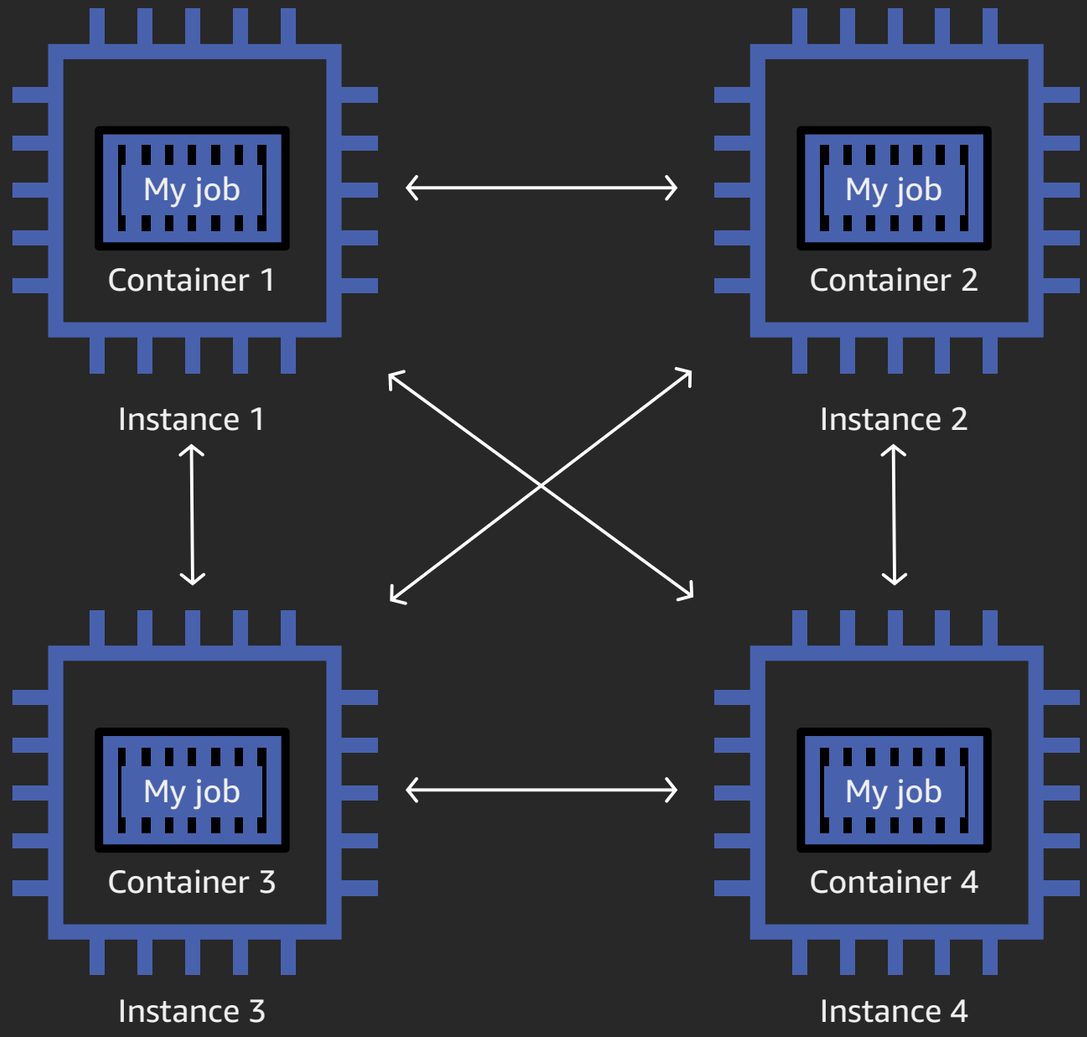
Multi-node parallel jobs on AWS Batch

A Multi-node parallel job enables AWS Batch to run single jobs which span multiple Amazon EC2 instances

Think: Distributed deep learning, HPC

Integrated with the Elastic Fabric Adapter (EFA) for low latency between nodes

All-or-nothing scaling



GPU scheduling

- We accept jobs based on the number of GPUs they require and scale up

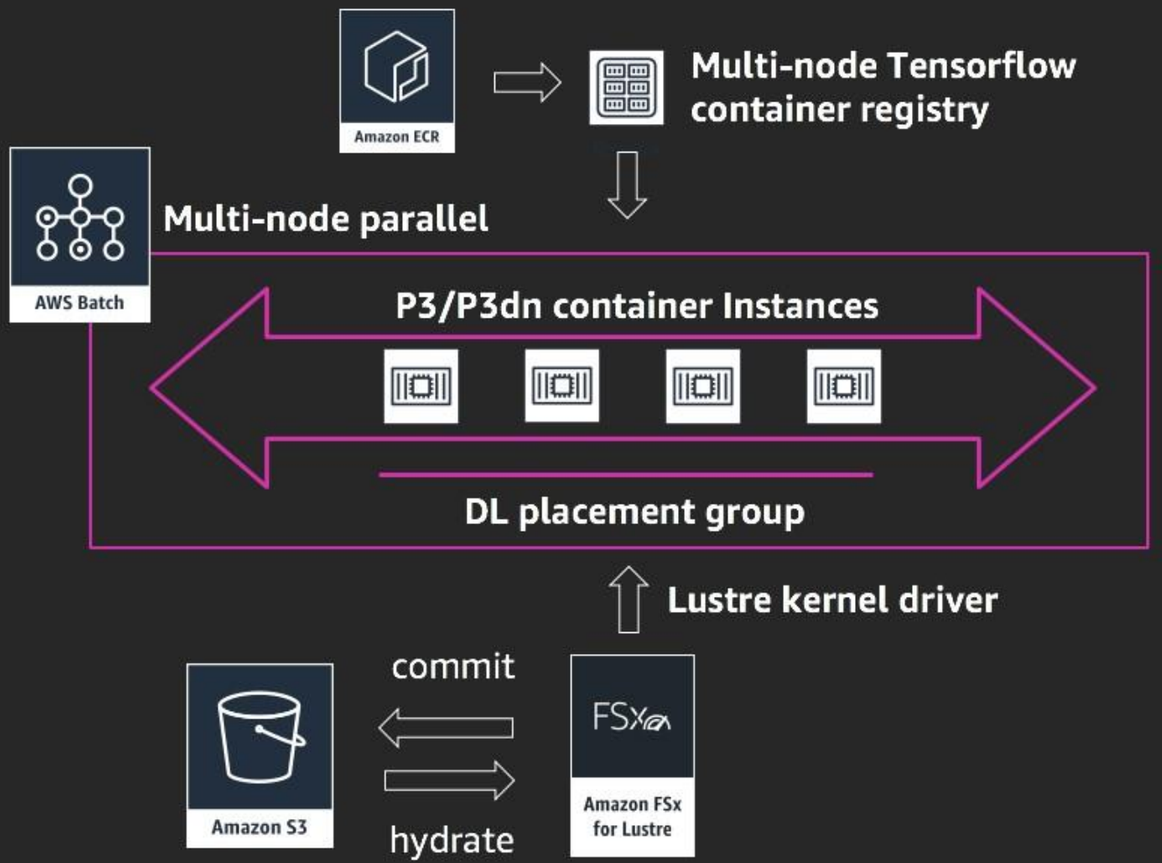
  P- or G-family instances accordingly

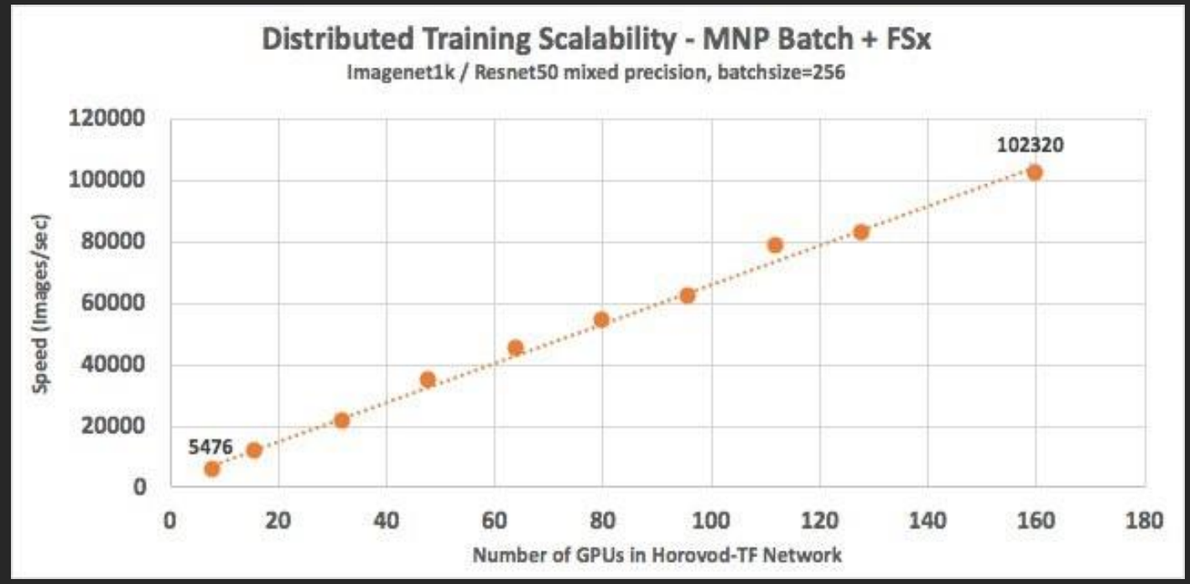
- We pin the GPU to the container to prevent oversubscription

- Will adopt new accelerators/GPUs as they are released

- Support for NVLink

Deep neural network training





AWS Batch is a natural fit for workflow managers

AWS Batch's sophisticated dependency capabilities and managed retries (yes, even on Spot termination) makes it an easy choice to build workflow managers

Better to eat the elephant one piece at a time than all at once, after all

The Uber ATG use case

Side- and rear-facing cameras work in collaboration to construct a continuous view of the vehicle's surroundings

Top-mounted lidar provides a 360° three-dimensional scan of the environment

Forward-facing camera array focusing both near and far field, watching for braking vehicles, crossing pedestrians, traffic lights, and signage

Roof mounted antenna provide **GPS positioning and wireless data capabilities**

Automatic Emergency Braking system operates independently as a safeguard for certain situations requiring activation of the vehicle braking system

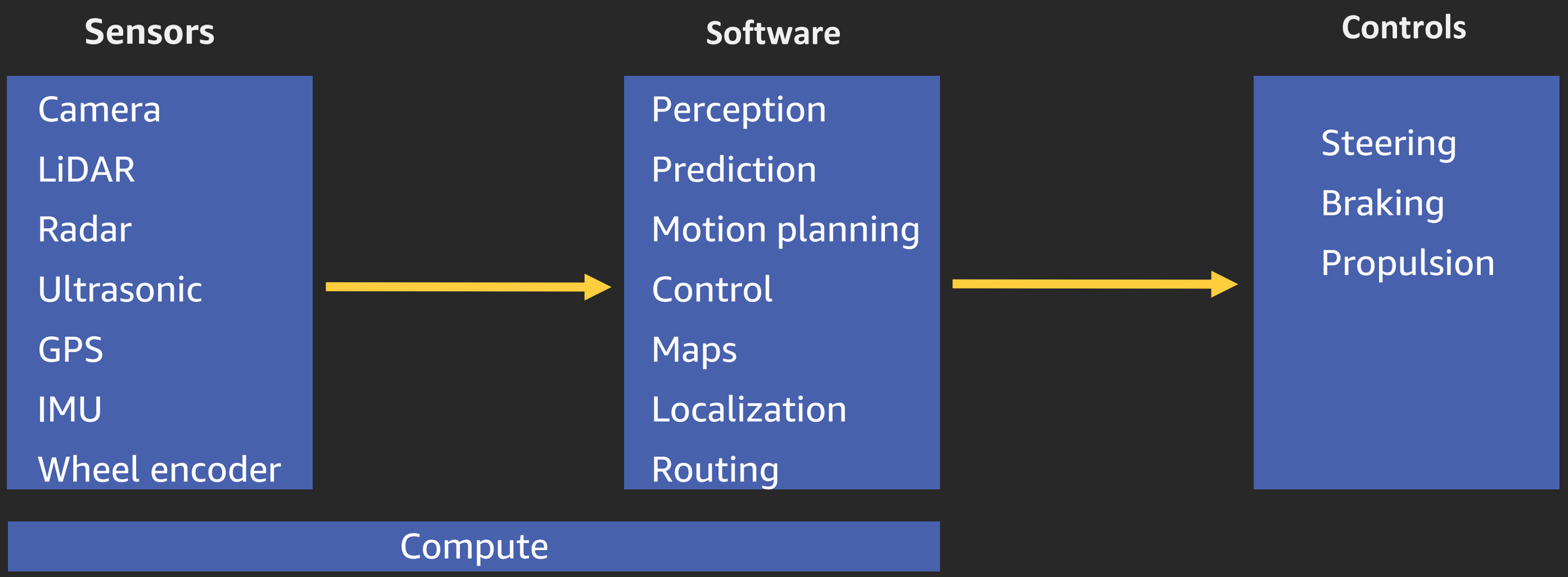
360° radar coverage detects vehicles and other obstacles

Custom designed compute and storage allow for real-time processing of data while a fully integrated cooling solution keeps components running optimally

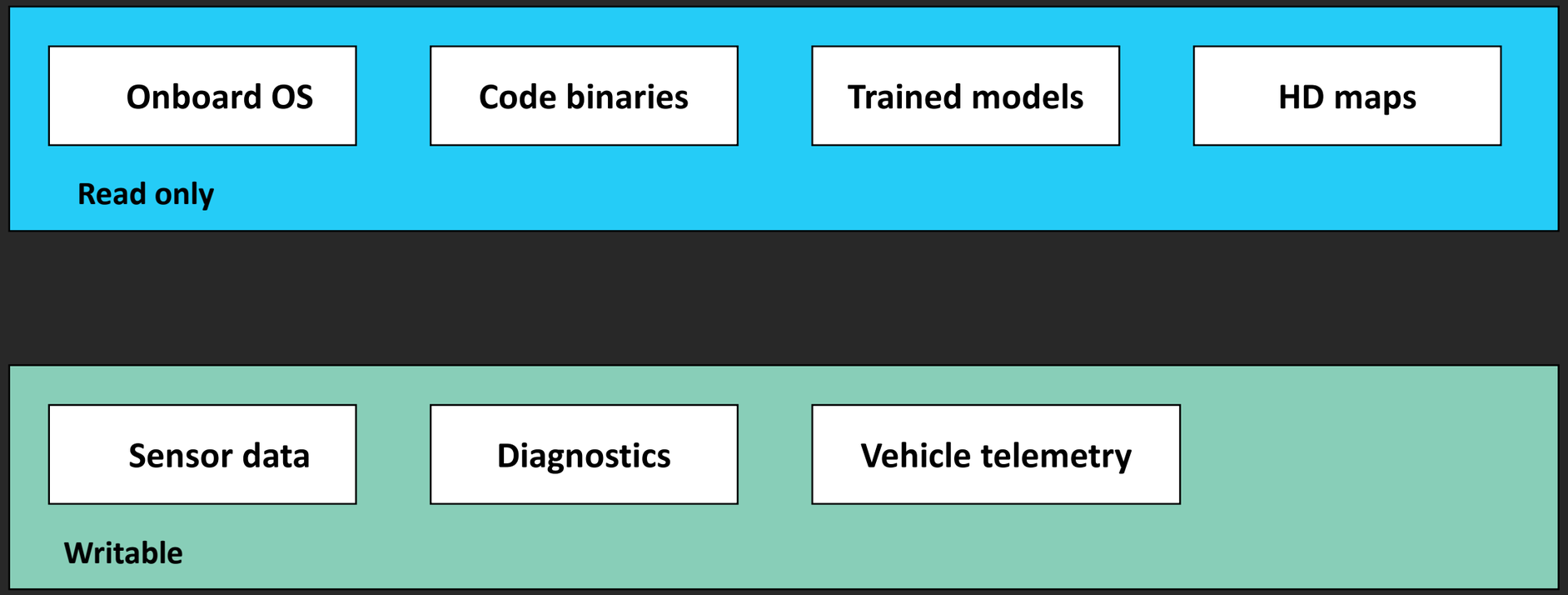
Gateway Module serves as a gateway to the base Vehicle Platform from the Uber Self Driving System, translating messages and commanding the vehicle's actuators (brakes, throttle, steering)

Modified base Vehicle Platform with Uber-specific mounting provisions, electrical harness, cooling interface, interior trim, and software control API

Self-driving vehicle basics



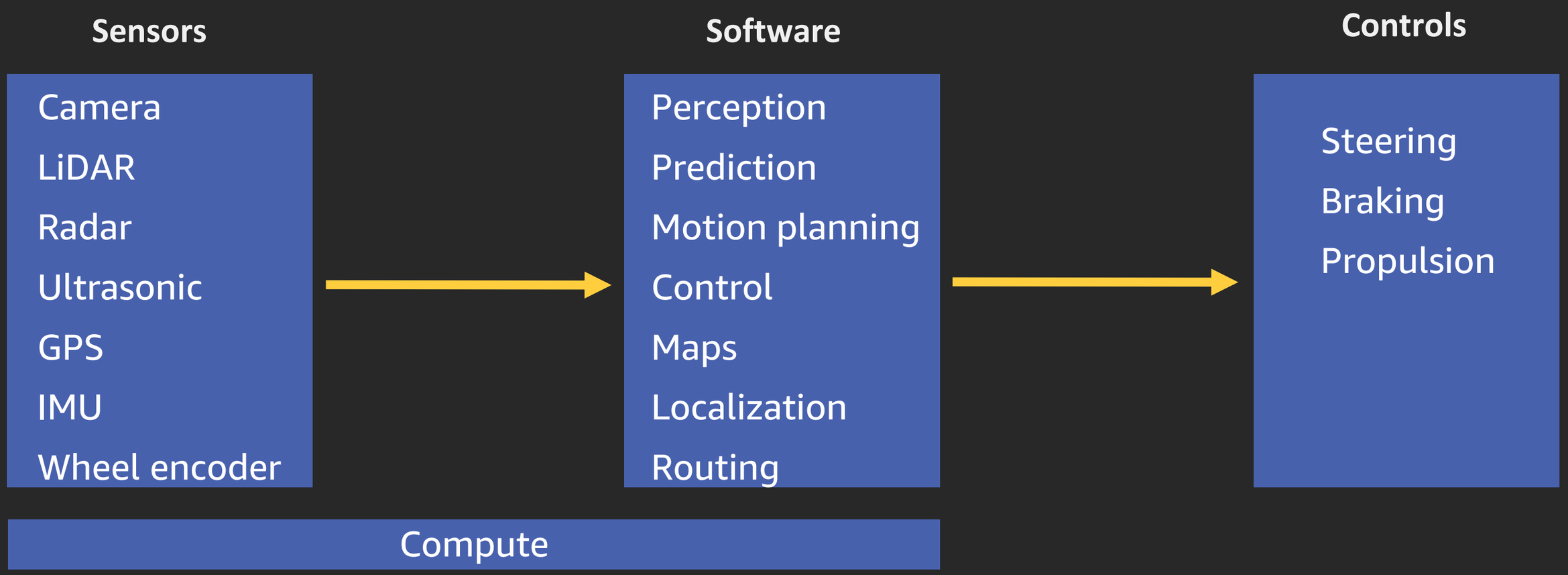
Onboard data



Testing

- Unit tests
- Sanitizers: ASan, MSan, TSan, UBSan
- Integration tests
- Feedback cycles require end-to-end testing

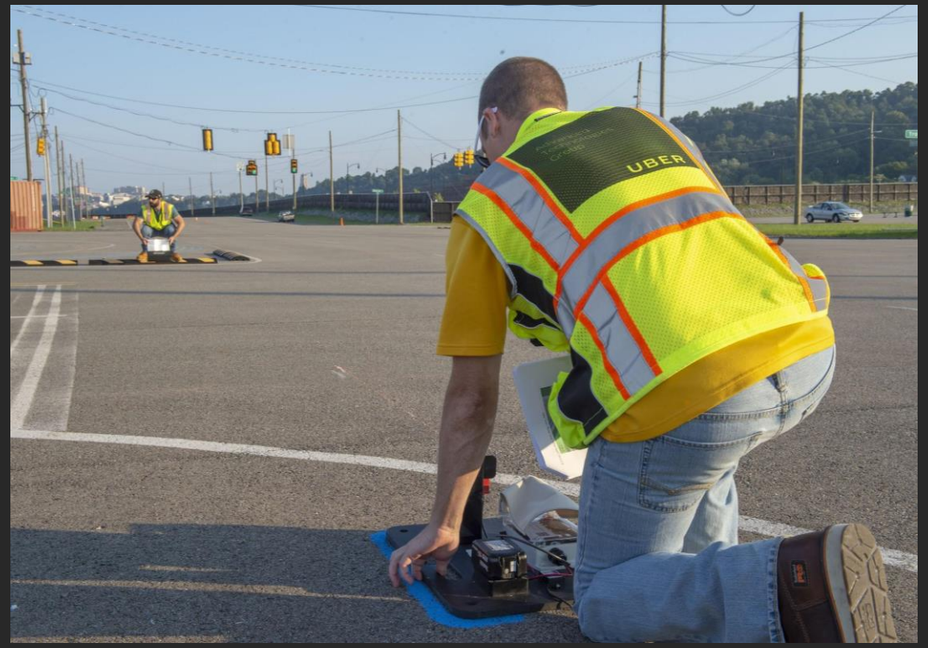
Self-driving vehicle basics



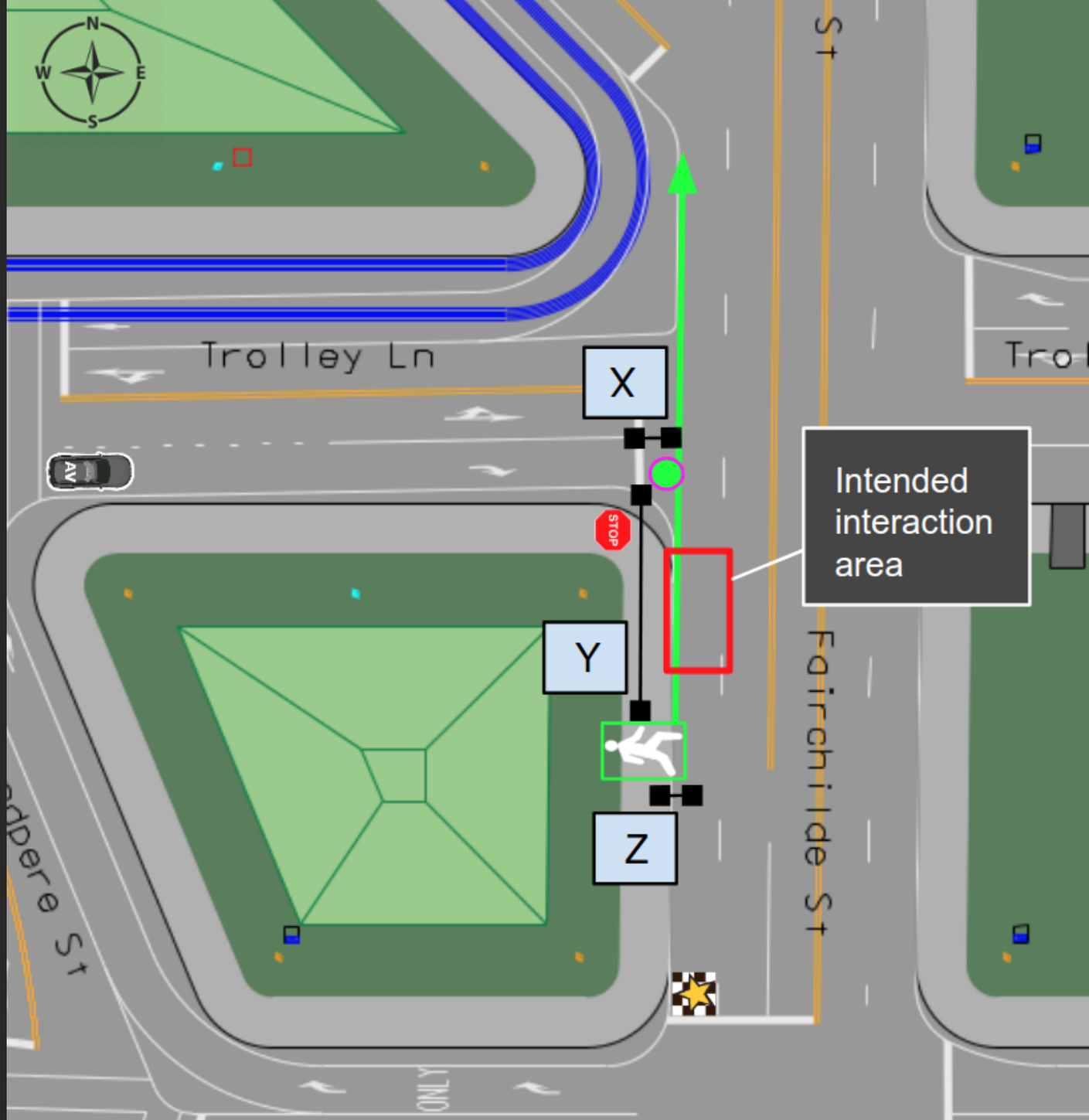

ONLY
ONLY


SCHOOL BUS
32
2028


UBER


UBER

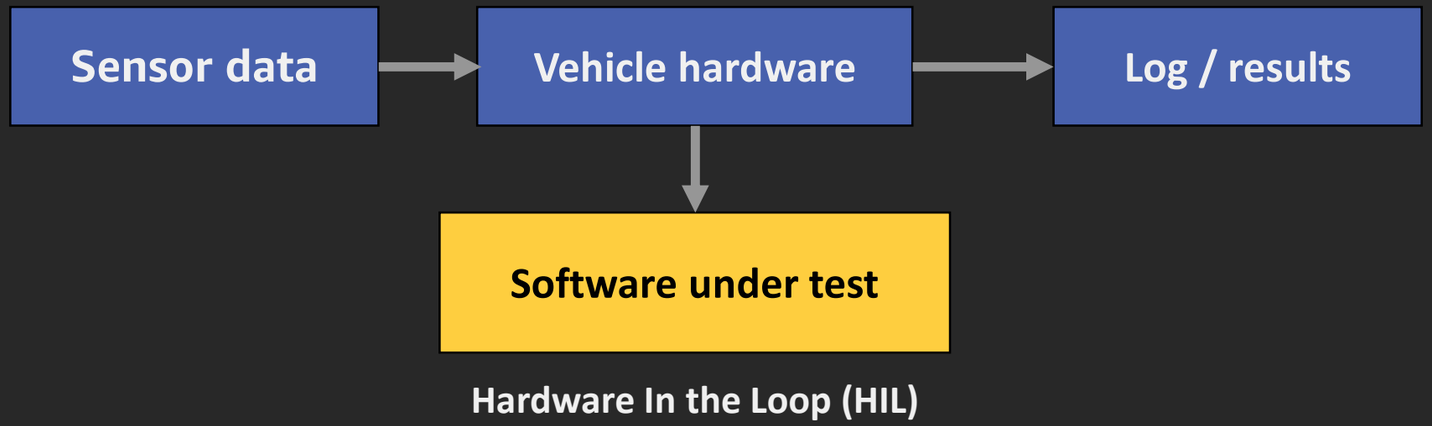
Scenario-based testing



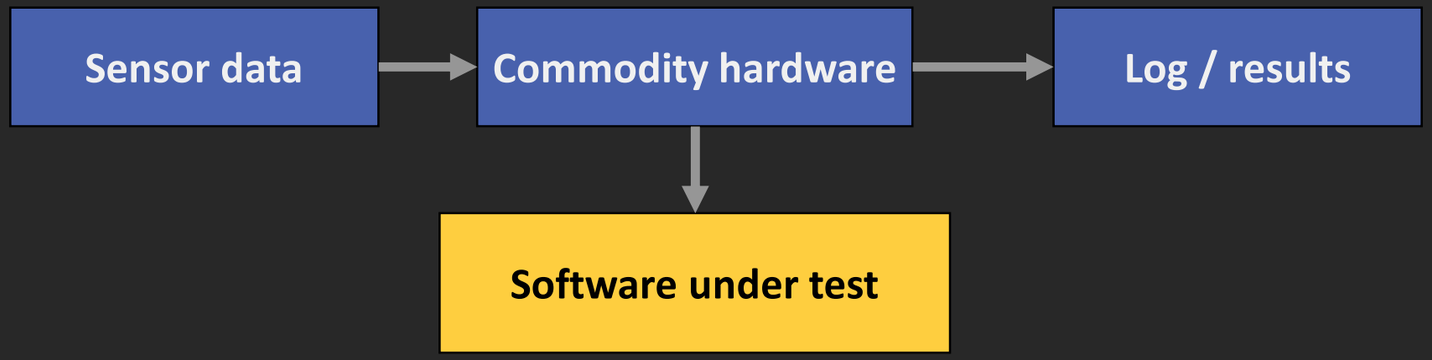
Track throughput

- Time intensive
- Space intensive
- Want to test each diff independently

Simulation

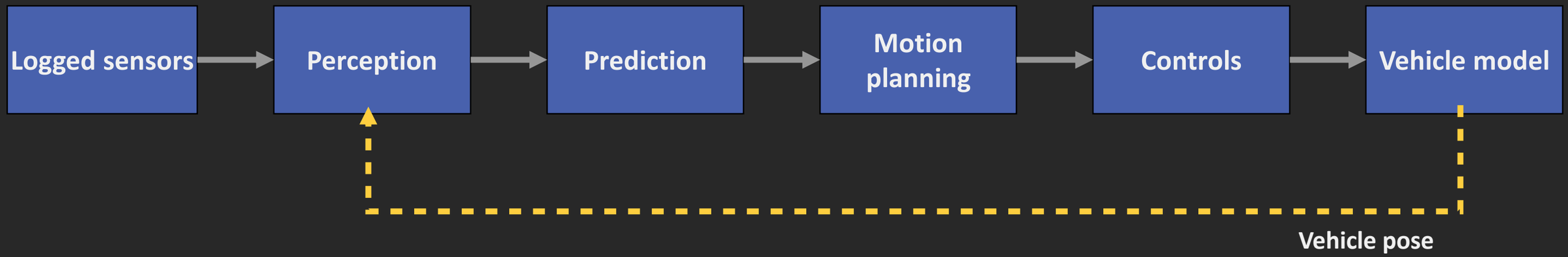


Hardware In the Loop (HIL)

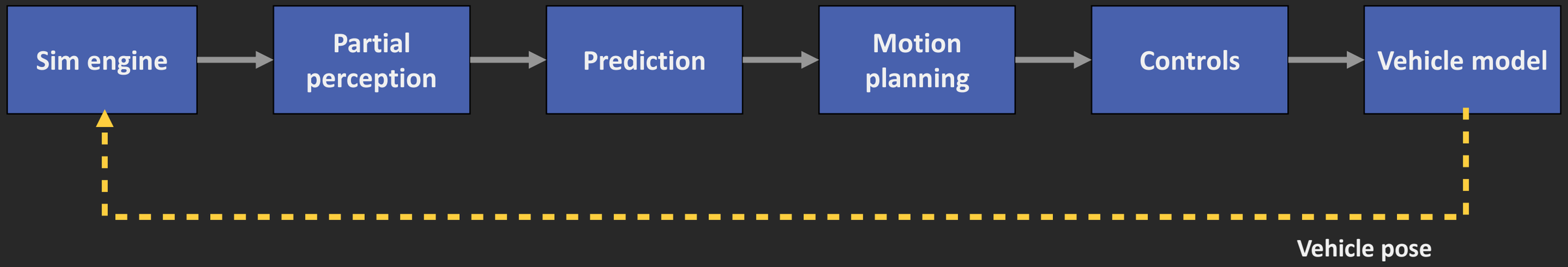


Software In the Loop (SIL)

Log-based simulation

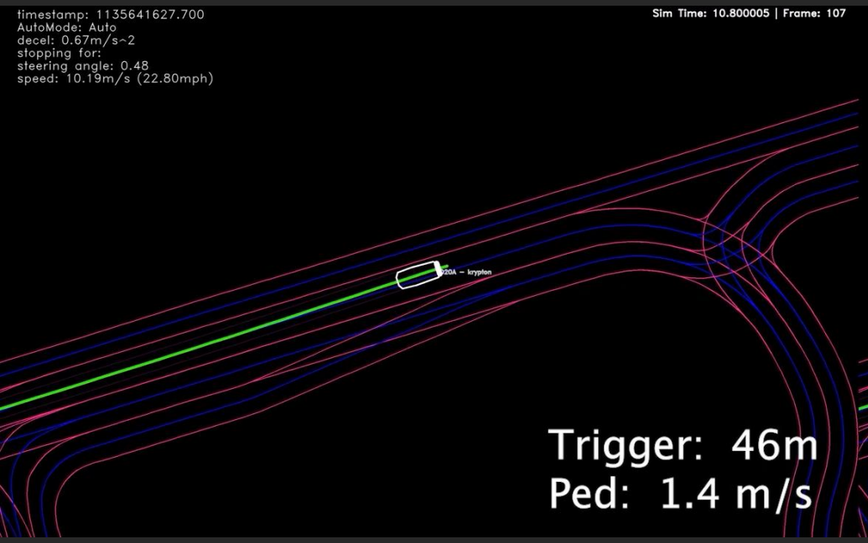


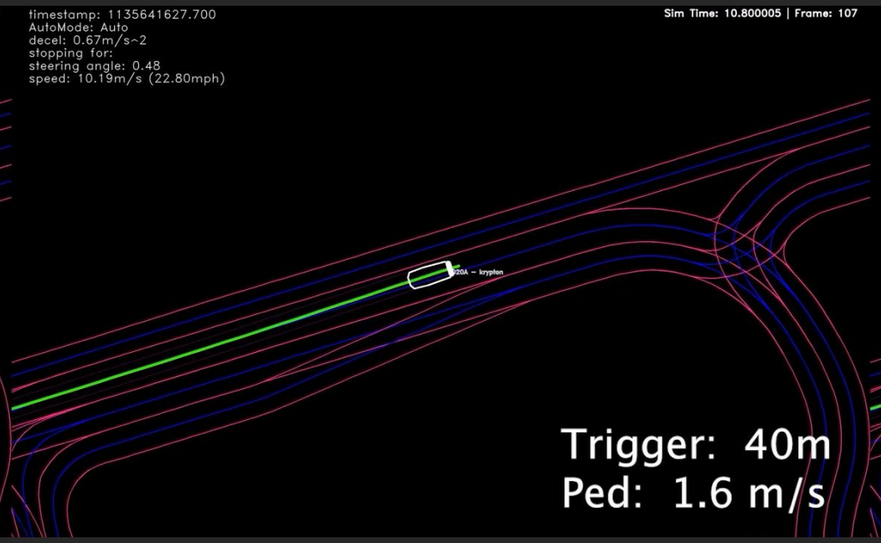
Virtual simulation

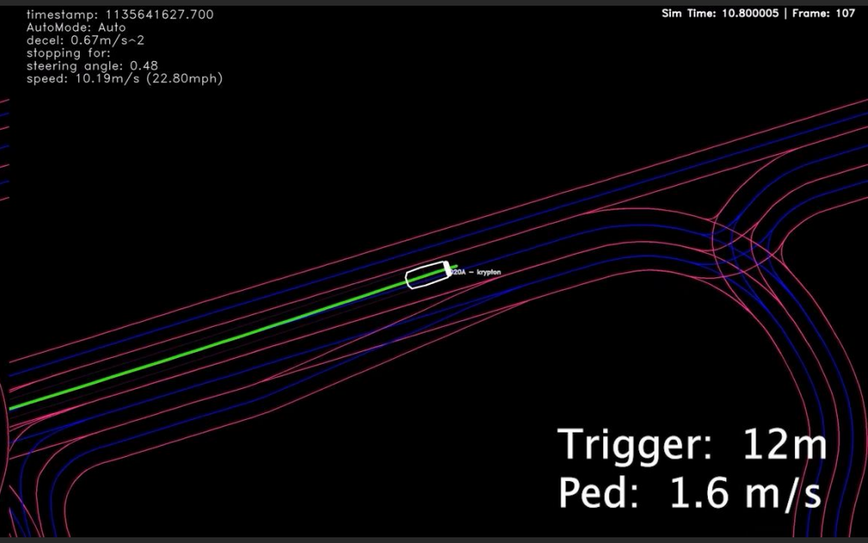


Variations

Variations


timestamp: 1135641627.700
AutoMode: Auto
decel: 0.67m/s^2
stopping for:
steering angle: 0.48
speed: 10.19m/s (22.80mph)
Sim Time: 10.800005 | Frame: 107
Trigger: 46m
Ped: 1.4 m/s


timestamp: 1135641627.700
AutoMode: Auto
decel: 0.67m/s^2
stopping for:
steering angle: 0.48
speed: 10.19m/s (22.80mph)
Sim Time: 10.800005 | Frame: 107
Trigger: 40m
Ped: 1.6 m/s


timestamp: 1135641627.700
AutoMode: Auto
decel: 0.67m/s^2
stopping for:
steering angle: 0.48
speed: 10.19m/s (22.80mph)
Sim Time: 10.800005 | Frame: 107
Trigger: 12m
Ped: 1.6 m/s

Variations Explorer

Scenario Selection

1. Please select a scenario

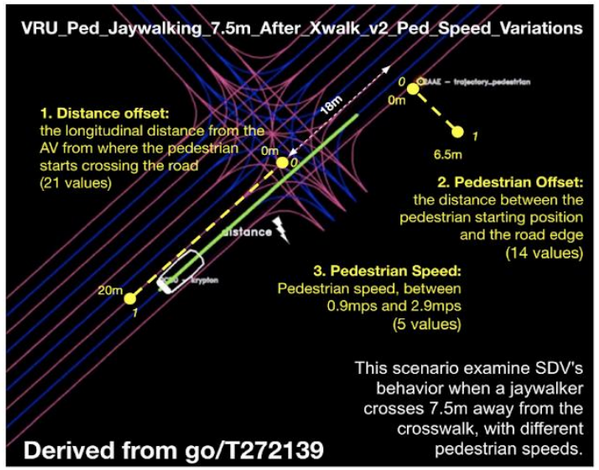
- By scenario id
- By scenario name

VRU_Ped_Jaywalking_7.5m_After_Xwalk_v2_Ped_Speed_Variations

You don't find the scenario you're looking for? Contact timothe@ to add it to Variations Explorer.

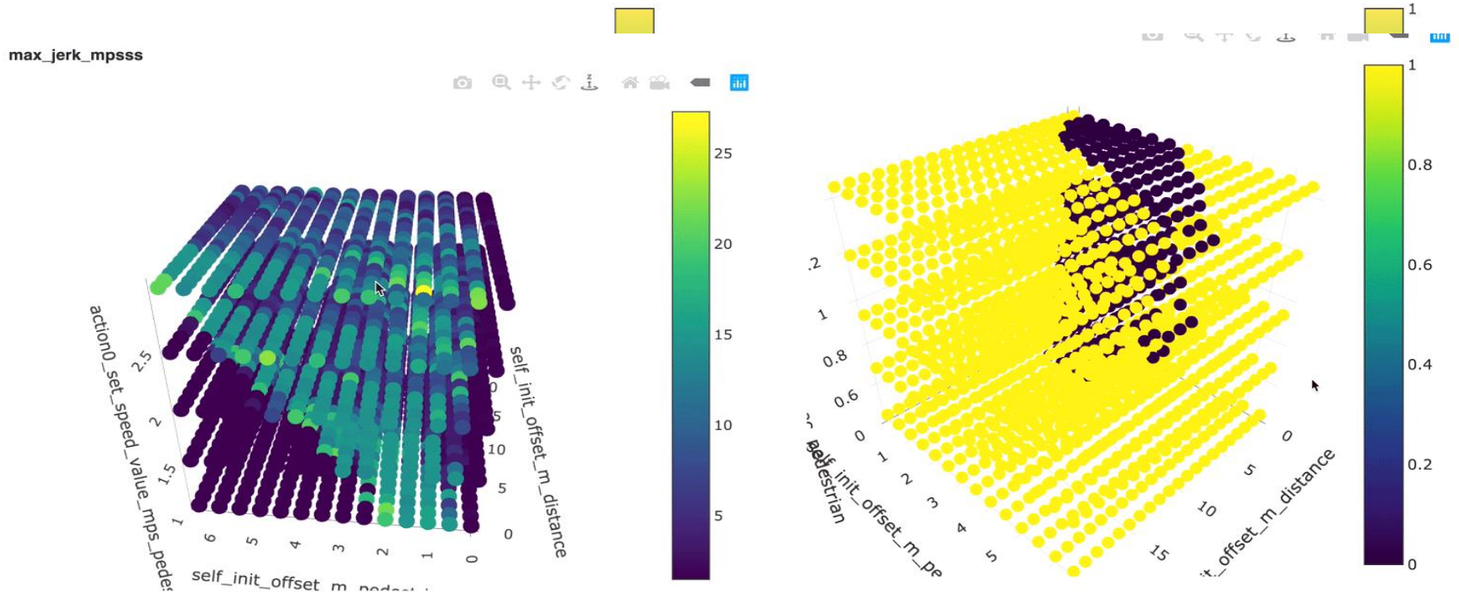
Scenario Specifications

Scenario Map



Scenario Identity Card:

Scenario name:
VRU_Ped_Jaywalking_7.5m_After_Xwalk_v2_Ped_Speed_Variations
Scenario id: a5e1568f-1008-0815-0021-01fec5396bc0
Number of parameters: 6
Best pass-fail rate over the past month: 99.59%
Average pass-fail rate over the past month: 83.64%
List of variation parameters:
'action1_accelerate_until_target_speed_mps_pedestrian',
'action0_set_speed_value_mps_pedestrian',
'self_init_offset_m_pedestrian'



Dot size

Color Scale

Green-Red | Colorblind-friendly colorscale | Viridis | Blues | Blue-Red
Electric | Greens | Rainbow | Red-Blue | Reds | Earth | Blackbody
Greys | Hot | Jet

Copy link to share analysis or click to reload the page

Clear everything and start from scratch

Side-by-side Base-Diff Comparison

This section allows you to compare 2 software releases and visualize new passes (NP), new failures (NF), common passes (CP), and common failures (CF). Please select 2 software releases in the section above. The X, Y and Z parameters are the same as those selected for the Matrix 1 (the left matrix).

Base (Left Matrix): B4792048-D197465
Diff (Right Matrix): B4792048-D197465


STOP

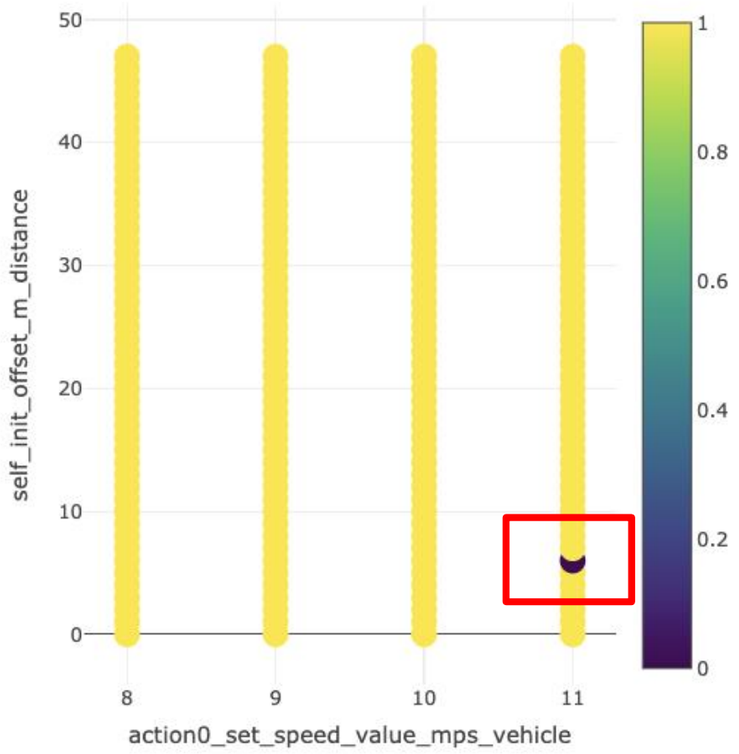
Scenario Map

Scenario Identity Card:

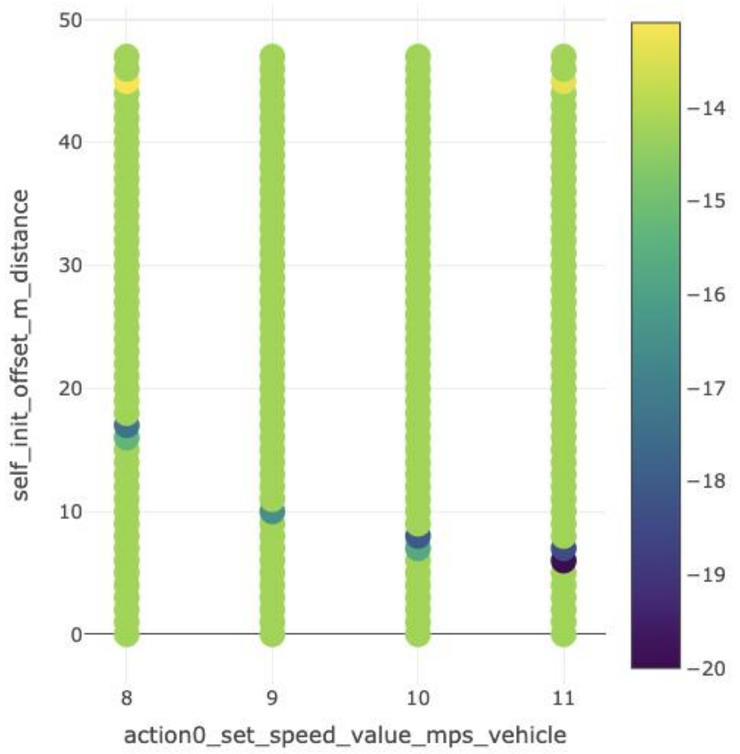
Scenario name:

N umberland_2way_SS_UPR_TestCase_Trigger_Variation_ACC_Off_S

pass-fail rate



min_jerk_mpsss



Search panels... (Ctrl+K)

Default

Log Info | Metrics | Settings | Debug

00:12 00:17 00:22 00:27 00:32 00:37

Acceleration

0
-2

Autonomous

1.0
0.5
0.0

Faulted

0.0
-0.5
-1.0

Velocity (SDV)

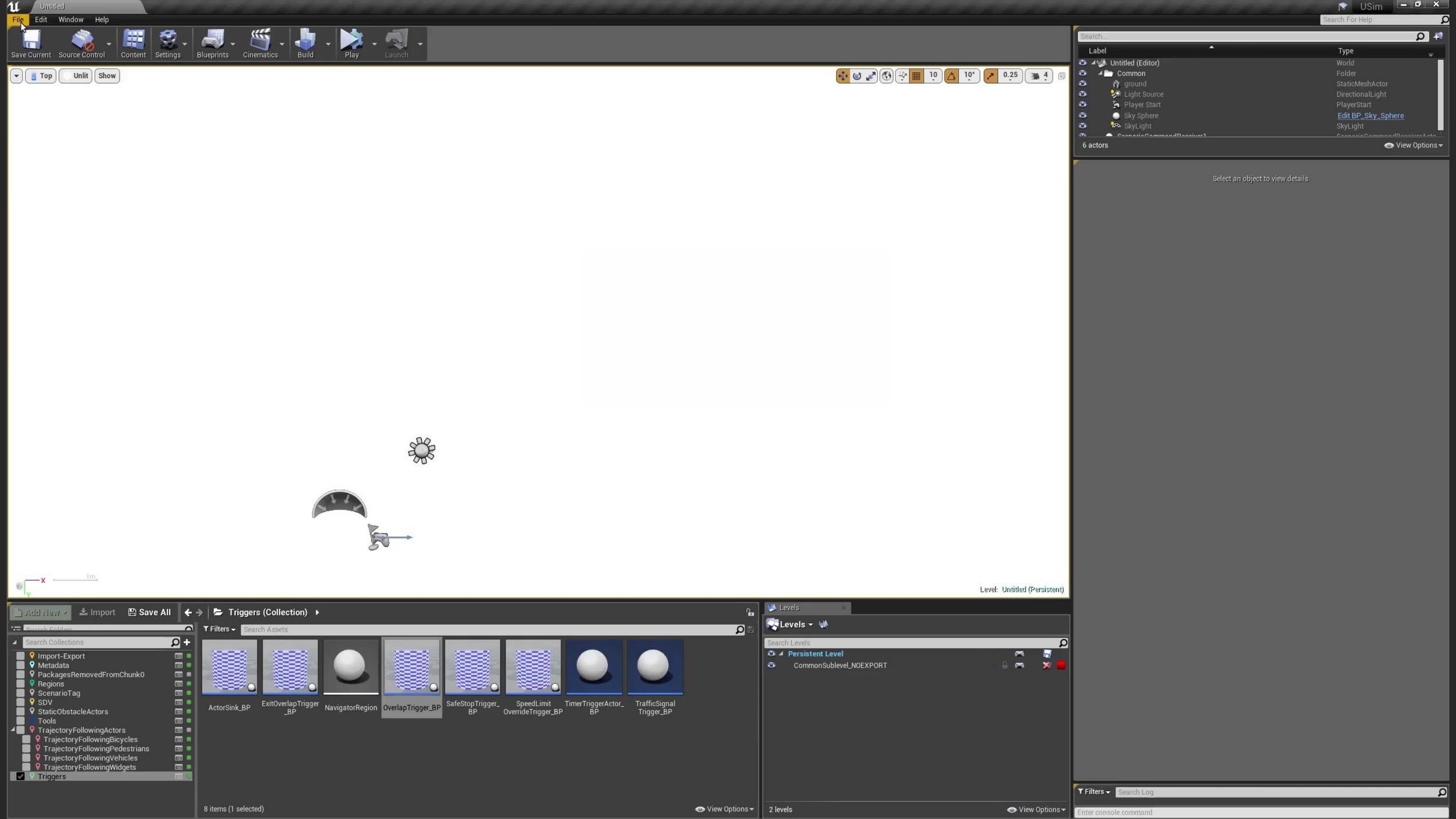
8
6
4
2
0

Ready

1.0
0.5
0.0

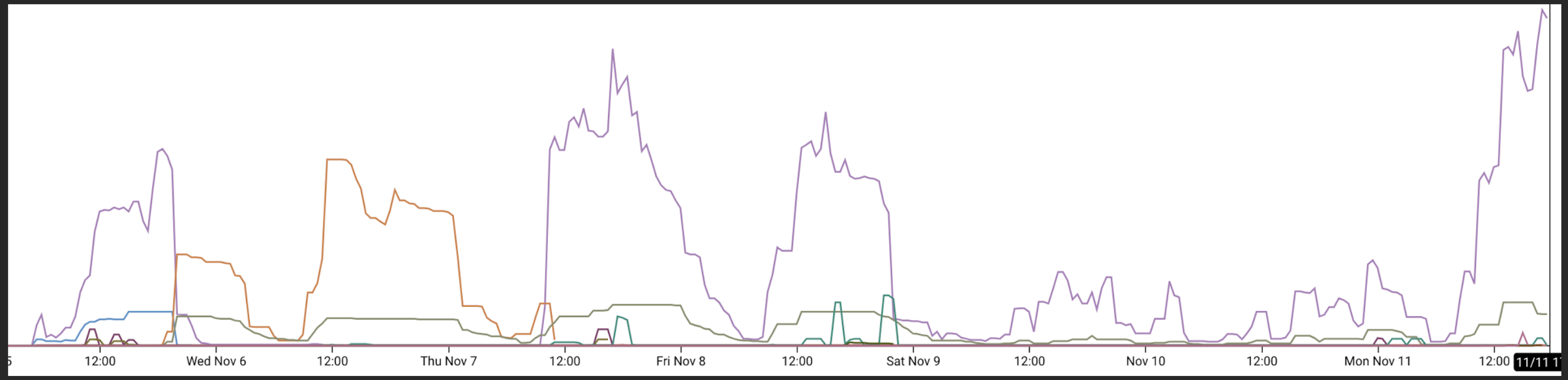

5226


Turn left
DELL

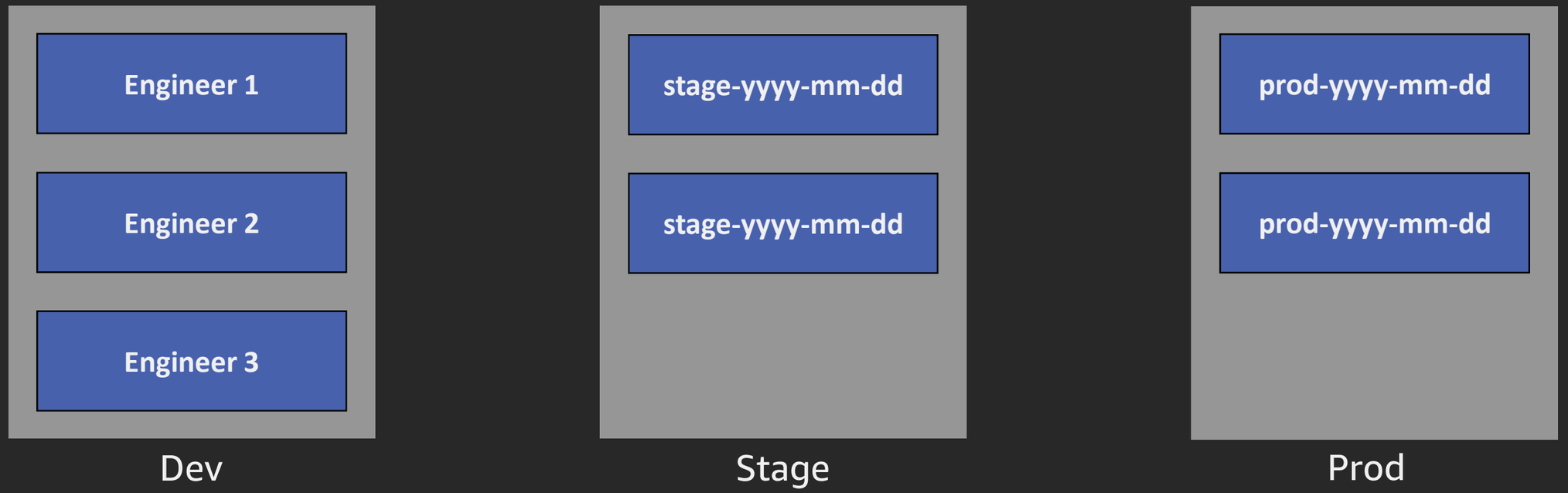

Untitled
USim
File
Edit
Window
Help
Search For Help
Save Current
Source Control
Content
Settings
Blueprints
Cinematics
Build
Play
Launch
Top
Unlit
Show
Label
Type
Untitled (Editor)
World
Common
Folder
ground
StaticMeshActor
Light Source
DirectionalLight
Player Start
PlayerStart
Sky Sphere
Edit BP_Sky_Sphere
SkyLight
SkyLight
6 actors
View Options
Select an object to view details.
Level: Untitled (Persistent)
Add New
Import
Save All
Triggers (Collection)
Filters
Search Assets
Search Collections
Import-Export
Metadata
PackagesRemovedFromChunk0
Regions
ScenarioTag
SDV
StaticObstacleActors
Tools
TrajectoryFollowingActors
TrajectoryFollowingBicycles
TrajectoryFollowingPedestrians
TrajectoryFollowingVehicles
TrajectoryFollowingWidgets
Triggers
ActorSink_BP
ExitOverlapTrigger_BP
NavigatorRegion
OverlapTrigger_BP
SafeStopTrigger_BP
SpeedLimitOverrideTrigger_BP
TimerTriggerActor_BP
TrafficSignalTrigger_BP
8 items (1 selected)
View Options
Levels
Search Levels
Persistent Level
CommonSublevel_NOEXPORT
2 levels
View Options
Filters
Search Log
Enter console command

Simulation in AWS

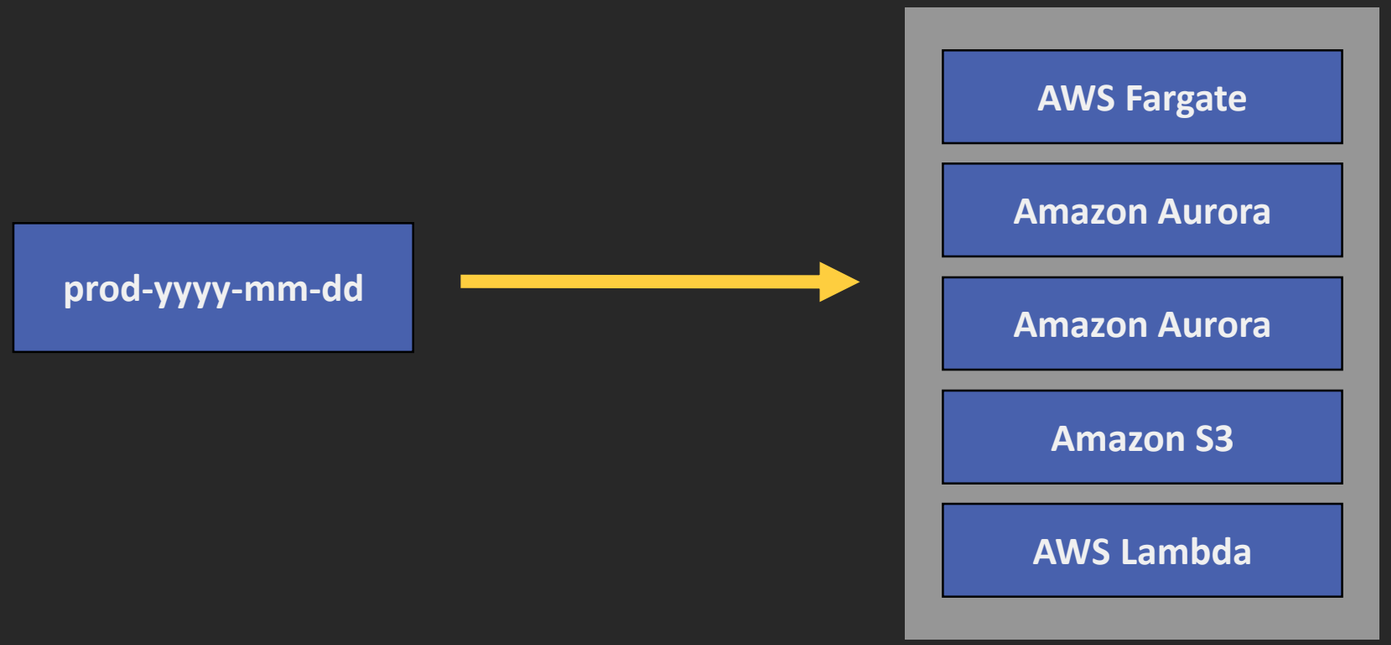
- This is an ideal workload for a public cloud
- Irregular demand
- Each test needs thousands virtual vehicles when running and zero when done
- Some experiments utilize 100K tests
- Some require 1M


12:00
Wed Nov 6
12:00
Thu Nov 7
12:00
Fri Nov 8
12:00
Sat Nov 9
12:00
Nov 10
12:00
Mon Nov 11
12:00
11/11

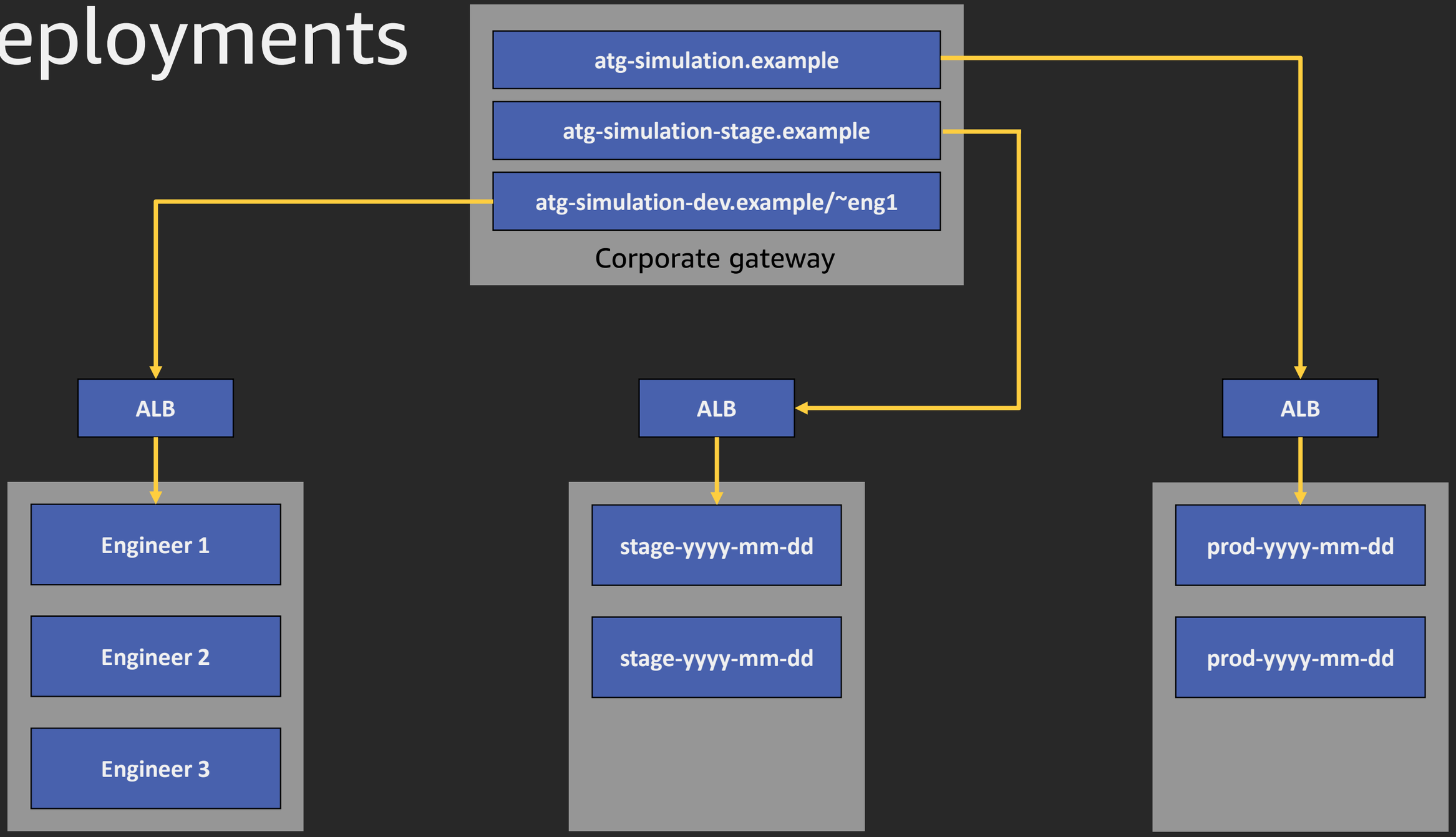
Deployments



Deployments

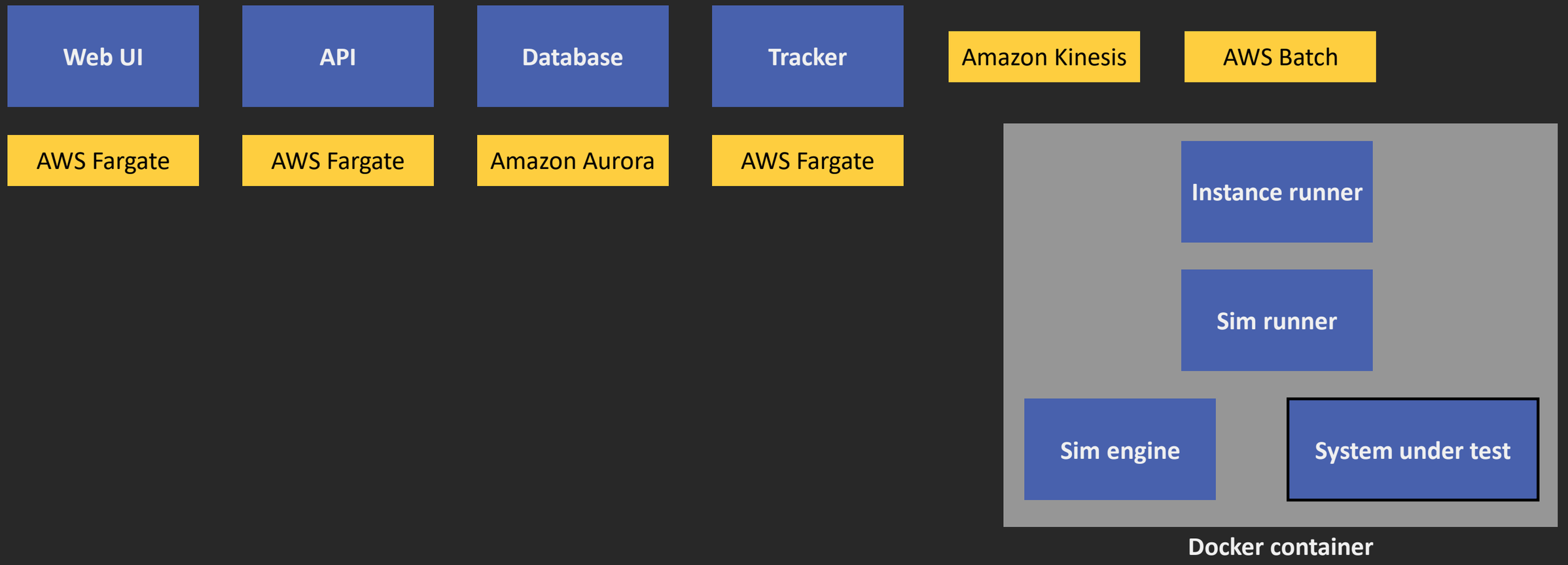


Deployments

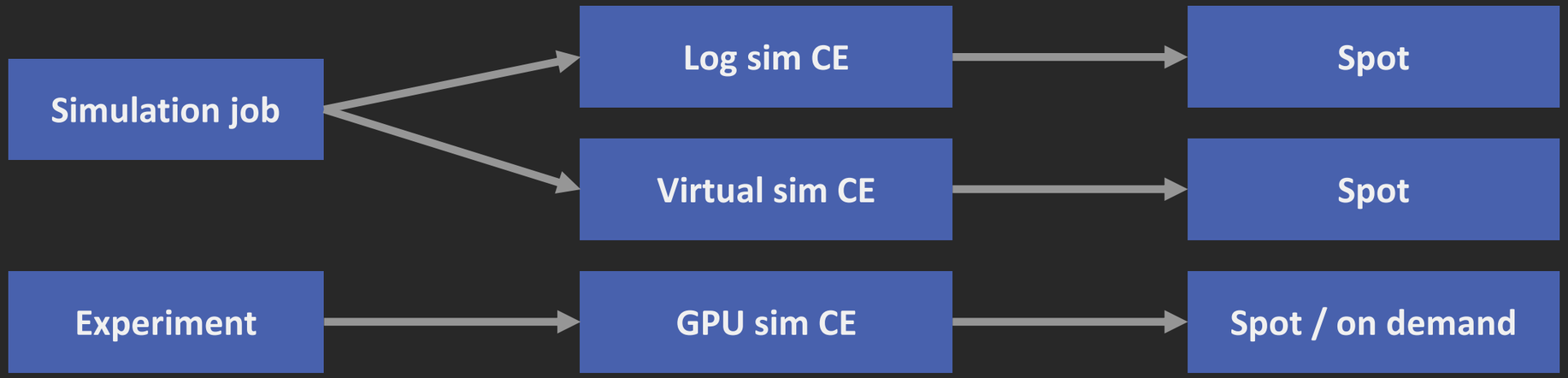


ATG components

ATG Components



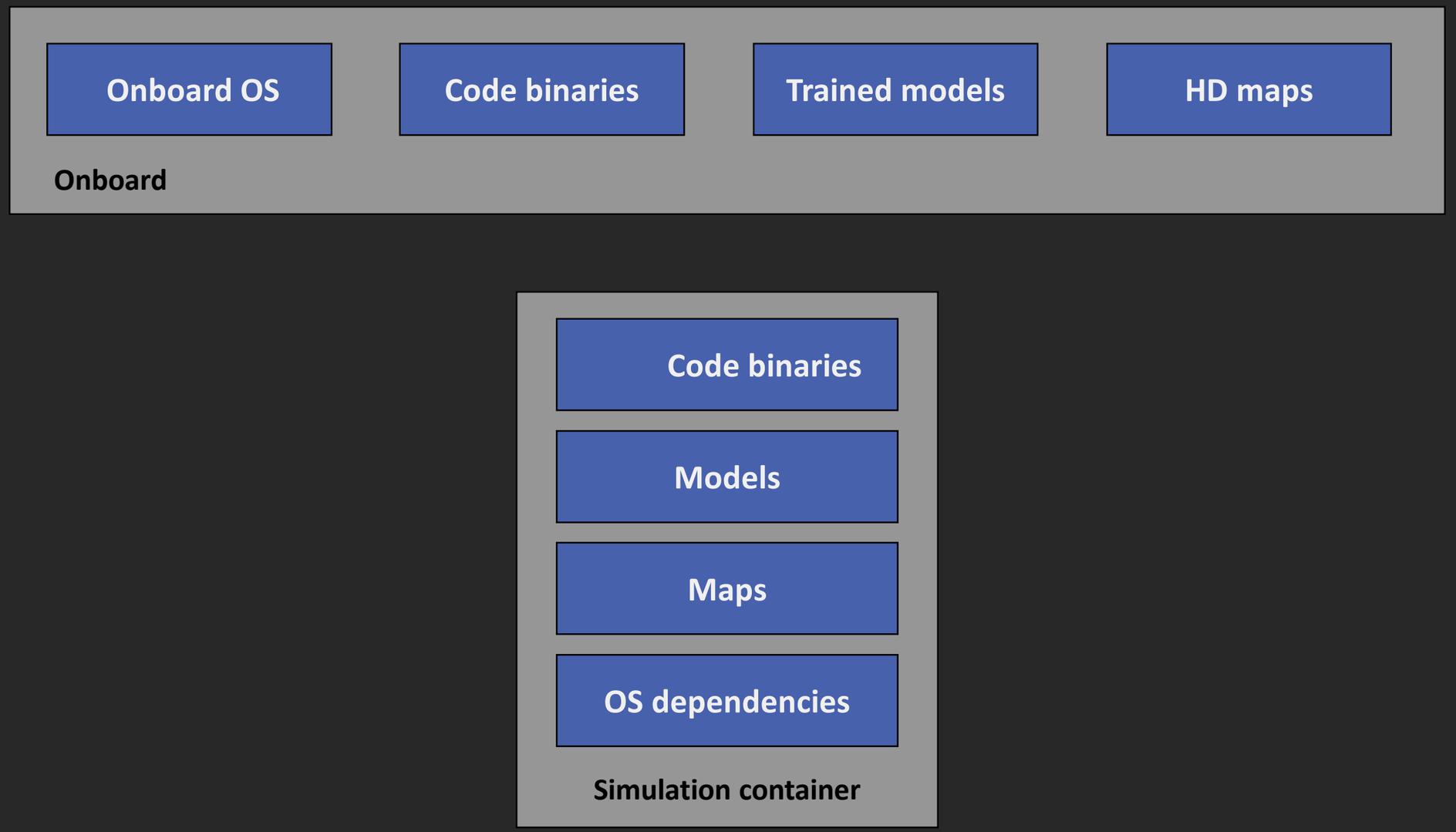
Compute environments



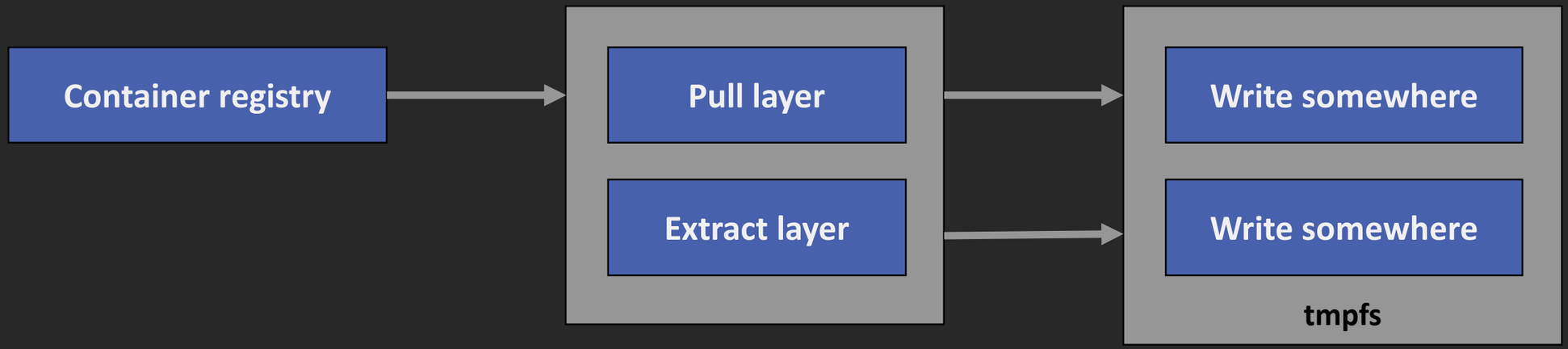
Why not use AWS Batch directly

- Containers are large

Large containers



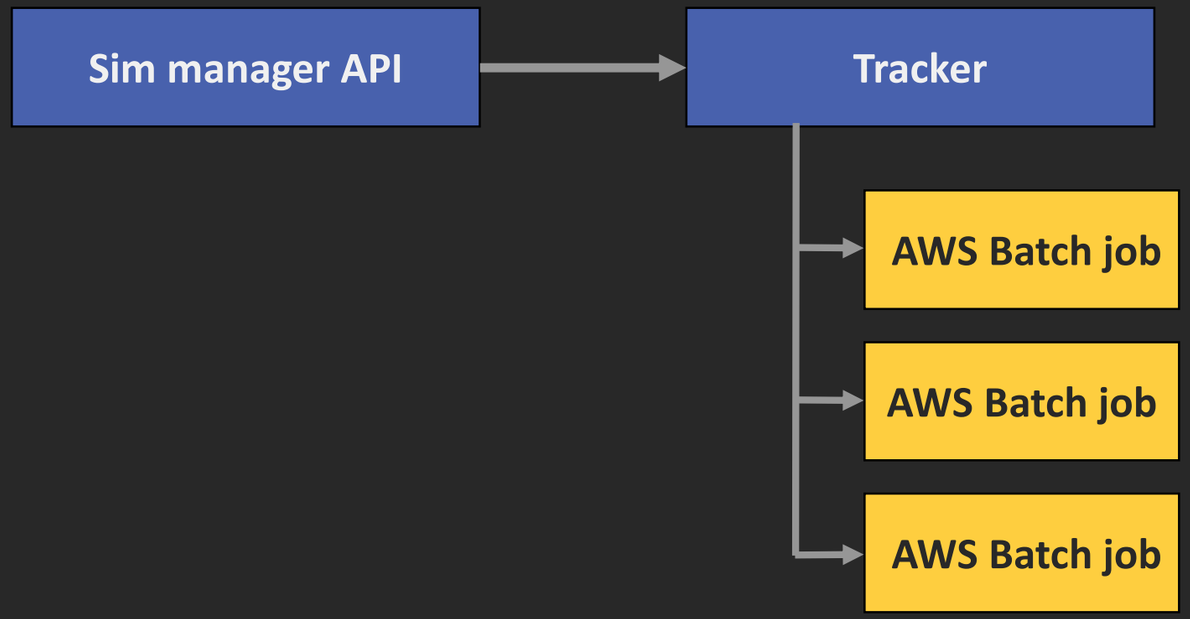
Write amplification



Why not use AWS Batch directly?

- Containers are large
- Job size limits

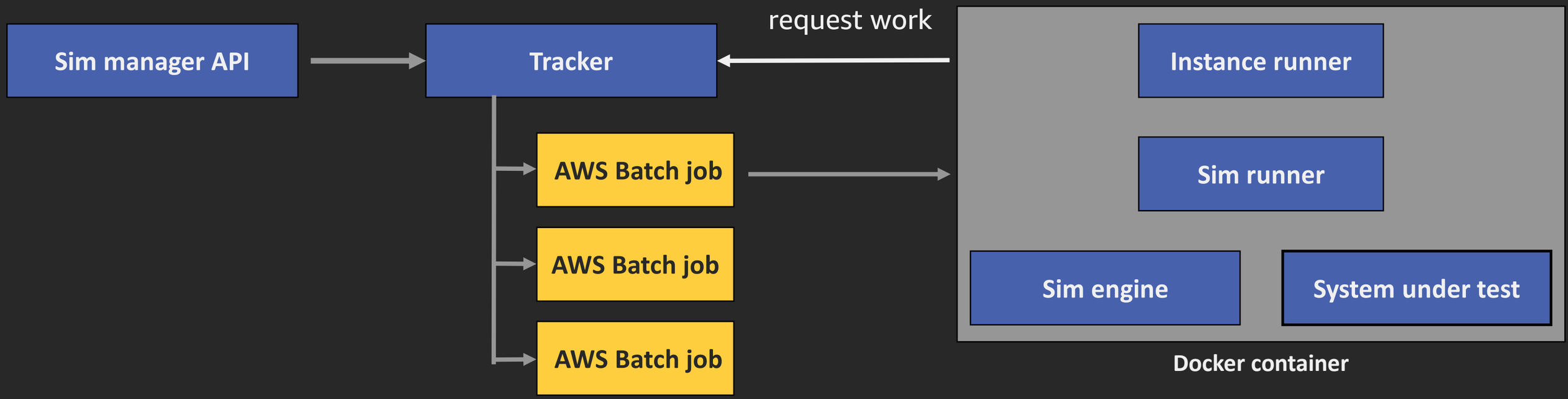
ATG Components



Why not use AWS Batch directly

- Containers are large
- Job size limits
- Smaller task lengths

ATG components



Why not use AWS Batch directly

- Containers are large
- Job size limits
- Smaller task lengths
- Observability

Observability

fcc9dcbd-62db-4a5b-ae91-ee17643d0bad

Variation Explorer (Beta) · Request for VE Notification

Created by: owner@example.com · **Created at:** 12/02/2019 at 3:41:04AM ET · **Version:** prod-2019-11-26
Image: sim_SIMBOT_2019.333.01_CI-B5003699_2019-12-02T08-31-39-UTC · **Eval Type:** rAV
Purpose: AV Release · **Autonomy Platform:** Xenon
Selection: sets (Autonomy Capabilities Progression Set)

151 Scenarios

54 Pass **97** Fail **0** Error

Run Status

- [ ] Complete (151)
- [ ] Running (0)
- [ ] Pending (0)
- [ ] Starting (0)

Result

- [ ] Pass (54)
- [ ] Fail (97)
- [ ] Error (0)

Refine By

Name, id, result, etc.

Tags

This evaluation was run with custom options

Show Details

Scenarios Sets

Fail Rate	Scenario Name	Pass	Fail	Error	Total
0%	AC_PLD_Ped_Emerges_From_Car_TestCase Executed: v2 Published: v2 Draft/In QA: --	1	0	0	1
0%	AC_PLD_Blocking_Car_Moves_AMO_Variations Executed: v5 Published: v5 Draft/In QA: --	24	0	0	24
50%	AC_PLD_Lane_Sharing_Bus_Marlborough Executed: v1 Published: v1 Draft/In QA: v2	36	36	0	72
0%	AC_PLD_Type_3_S2_Bike_Following_Marlborough Executed: v2 Published: v2 Draft/In QA: --	6	0	0	6
100%	PD_1_1_No_Taper_Single_Cone_Truck_Backup_Min_Encroach - PD_1_1_No_Taper_Single_Cone_Truck_Backup_Min_Encroach Executed: v4 Published: v4 Draft/In QA: --	0	1	0	1
0%	AC_PLD_Ped_Emerges_From_Car Executed: v1 Published: v1 Draft/In QA: --	3	0	0	3
100%	PD_0_3_Nominal_Truck_Occlusion_Right_Turn_Blocked Executed: v7 Published: v7 Draft/In QA: --	0	1	0	1
0%	AC_PLD_Occluded_Oncoming_Car_UPL_TestCase	1	0	0	1

Back to Evaluation Summary

Metadata for **fcc9dcbd-62db-4a5b-ae91-ee17643d0bad**

Evaluation Summary

Evaluation ID	fcc9dcbd-62db-4a5b-ae91-ee17643d0bad
Evaluation Type	rav
Launching user	user@example.com
Launch time	2019-12-02T08:41:04.682Z
Purpose	av release
Purpose Description	
Exit Reason	success
Exit Reason Description	
Number of variations	682
Selected Tags	
Selected Scenarios	
Selected Sets	Autonomy Capabilities Progression Set
Docker image ID	276276014291.dkr.ecr.us-east-1.amazonaws.com/simulation-container:sim_SIMBOT_2019.333.01_CI-B5003699_2019-12-02T08-31-39-UTC
Evaluation S3 Path	evaluation/fcc9dcbd-62db-4a5b-ae91-ee17643d0bad
Kinesis stream	prod-2019-11-26_eval_fcc9dcbd-62db-4a5b-ae91-ee17643d0bad
Tracker cluster name	default

Observability

- Structured logging with JSON
- Works well with CloudWatch Logs
- Easily integrates with other log aggregators and JSON tools

Observability

Original log line:

```
[1135641617.000000] (route_map_interface_cache.cc:900) Route map interface cache: 0 / 0 nodes loaded,  0 / 629145600 bytes used (0 B / 600.000 MB)
```

Structured log line:

```
{
    "level": "info",
    "ts": "2006-01-02T15:04:05.132Z",
    "msg": "route map interface cache stats",
    "m:nodes_loaded": 0,
    "m:nodes_max": 0,
    "m:used_bytes": 0,
    "m:max_bytes”: 629145600,
}
```

Observability


```
{
    "level": "info",
    "ts": "2019-10-10T19:14:43.412Z",
    "caller": "awsutils/s3_downloader.go:70",
    "msg": "downloaded from s3",
    "deployable": "instance-runner",
    "deployment": "prod-2019-10-08",
    "environment": "prod",
    "evaluation_id": "f1a5e24cf-686c0-46db-a5ee-f5f896621234e",
    "compute_environment": "vsim-prod-spot-2019-09-20",
    "aws.private_ip": "10.104.20.68",
    "aws.region": "us-east-1",
    "aws.availability_zone": "us-east-1a",
    "aws.version": "2017-09-30",
    "aws.instance_id": "i-0a58501f1ad6e1aeb6",
    "aws.image_id": "ami-016b0f37bed512b2e2",
    "aws.instance_type": "c5.18xlarge",
    "batch_job_attempt": "1",
    "batch_job_index": "34",
    "batch_job_id": "709596b2-2fbc-43d6-bb18-b40a8bb72910:34",
    "bucket": "avmaps-mini-maps",
    "key": "14293243b6458ca2c5d6482bbe695b",
    "m:s3_downloaded_bytes": 42728717568
}
```

Select log group(s)

30m 1h 3h 12h custom (2d)

Clear /aws/batch/job

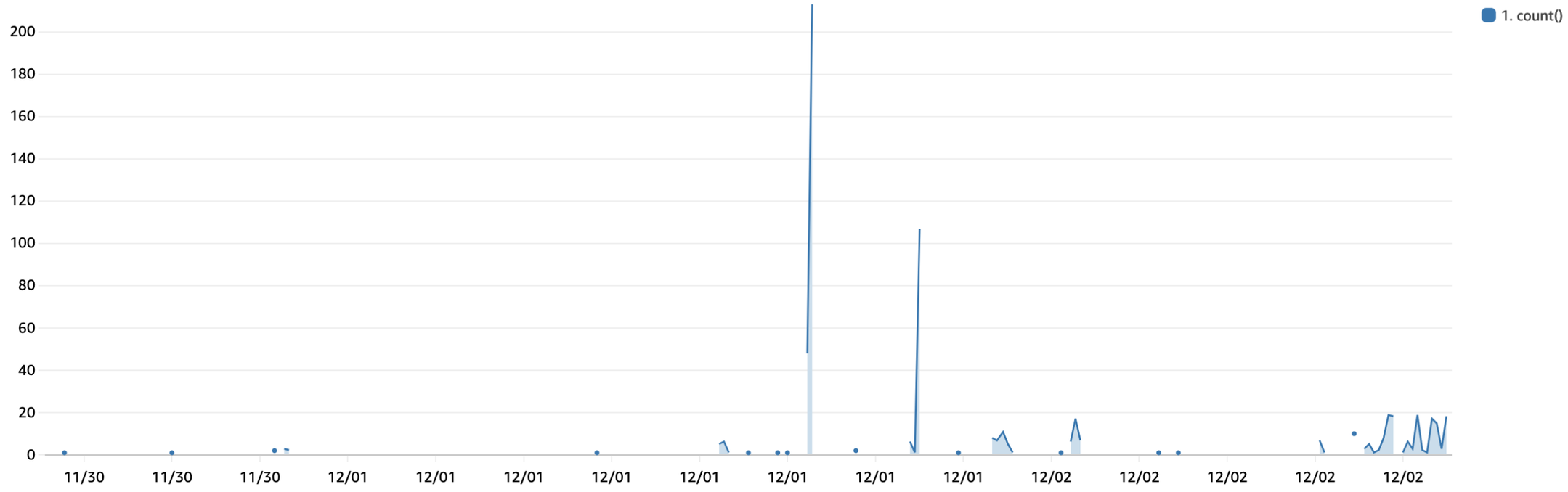
```
fields msg, attempt
| filter msg = "starting grpc server" and attempt > 0
| stats count() by bin(10min)
```

Run query History

Logs Visualization

Export results Add to dashboard

Stacked area



Challenges

- Large containers and write amplification
- Ephemeral storage on ephemeral compute
- Running out of memory
- AWS Batch is not very friendly for end users
- AWS Batch is getting faster all the time, but we still wish it was faster

We still use AWS Batch

- Handles our scale better than anything else we've tried
- Works well for experiments

Recommendations

- Work with account team to validate assumptions
- Keep Docker sizes small
- Add a pre-OOM killer
- Sometimes tmpfs is the best choice
- Log everything as JSON

AWS Batch roadmap

Roadmap

- Significant improvements to the AWS Batch console
- Full integration with container insights for job-level metrics
- Speed and scale (faster scaling, higher limits)
- Custom logging configuration
- New compute strategies and types
- Better placement logic and additional scheduling methods: Increase utilization, lower cost, increase scheduling fairness

And more

Thank you!

AWS re:Invent

aws

Please complete the session survey in the mobile app.