Enabling Reproducible and Agile Full-System Simulation

Work by Bobby R. Bruce, Ayaz Akram, Hoa Nguyen, Kyle Roarty, Mahyar Samani, Marjan Fariborz, Trivikram Reddy, Matthew D. Sinclair, and Jason Lowe-Power

THE UNIVERSITY OF CALIFORNIA DAVIS

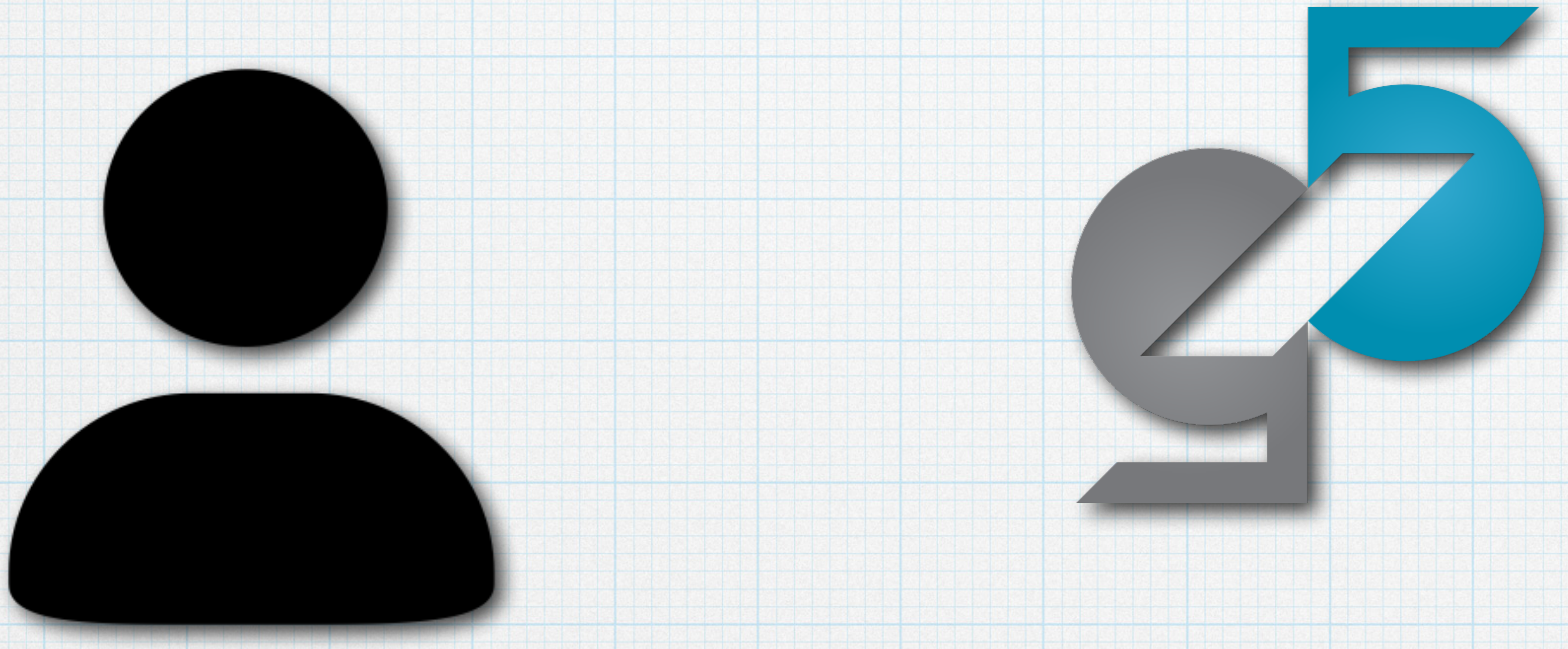
WISCONSIN
UNIVERSITY OF WISCONSIN-MADISON

AMD

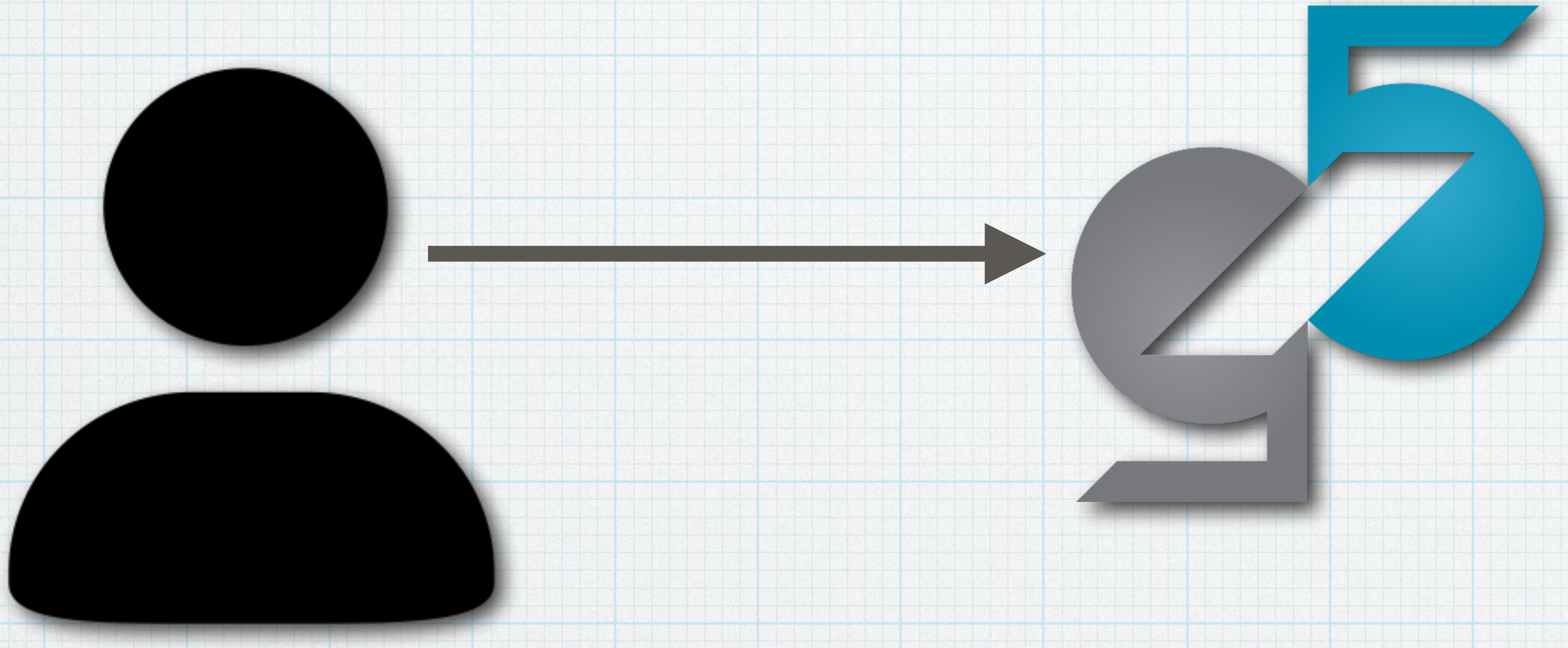
Presented by Bobby R. Bruce

The Problem

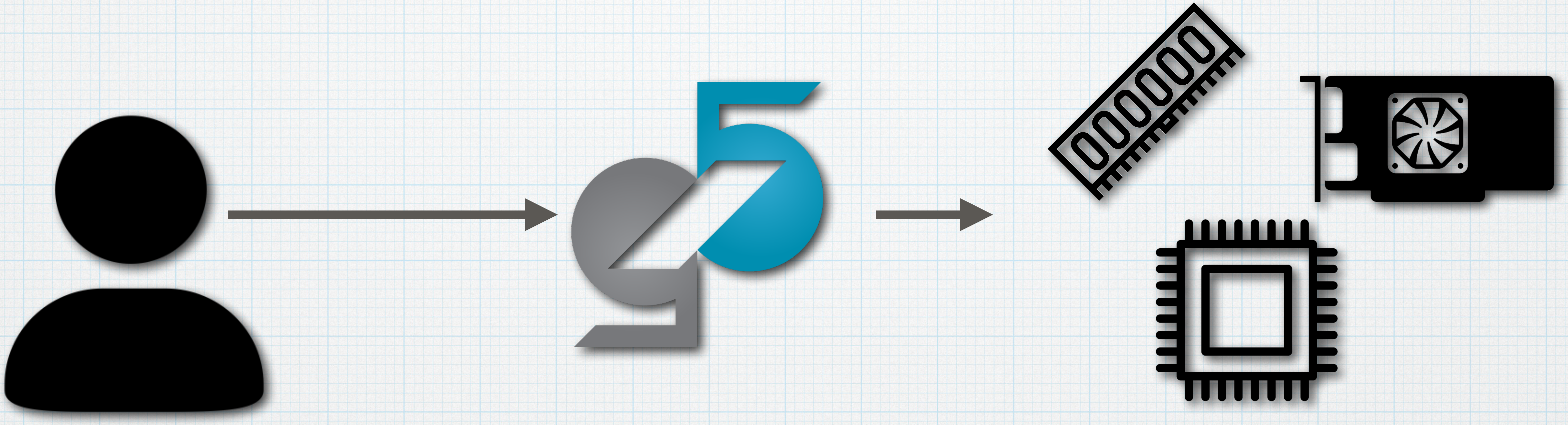
The Problem



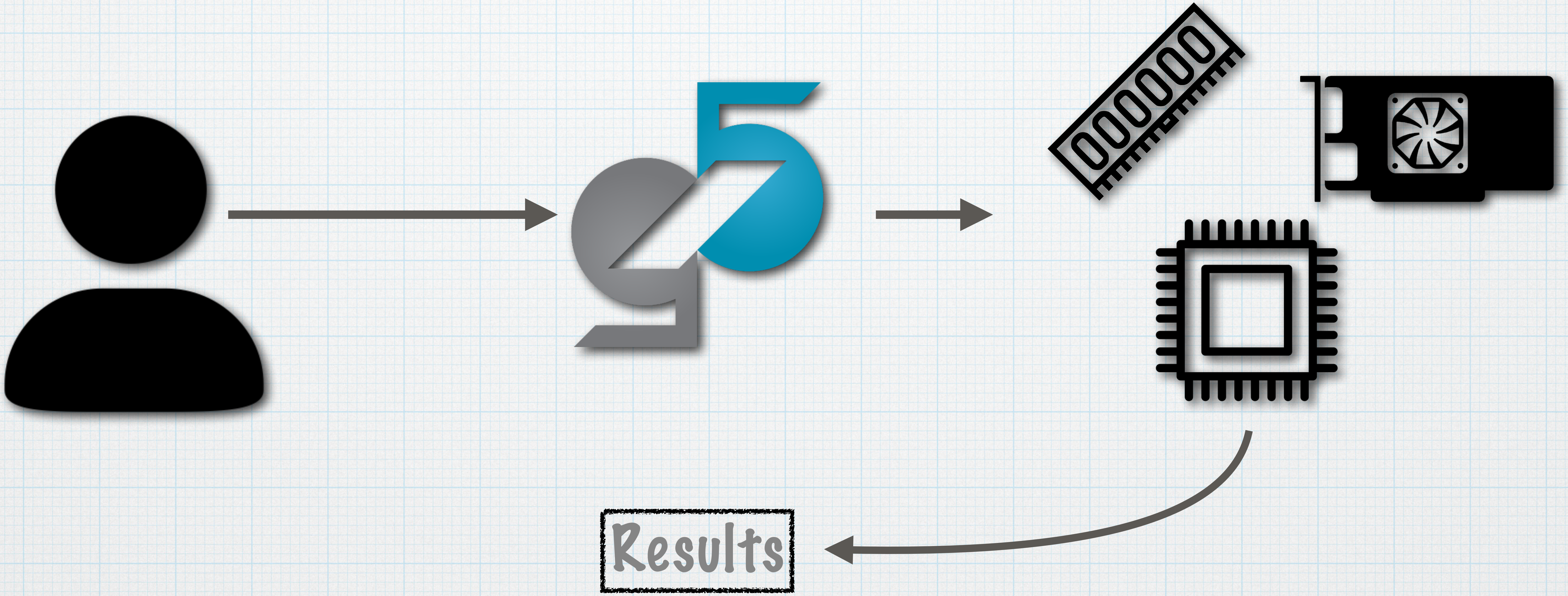
The Problem



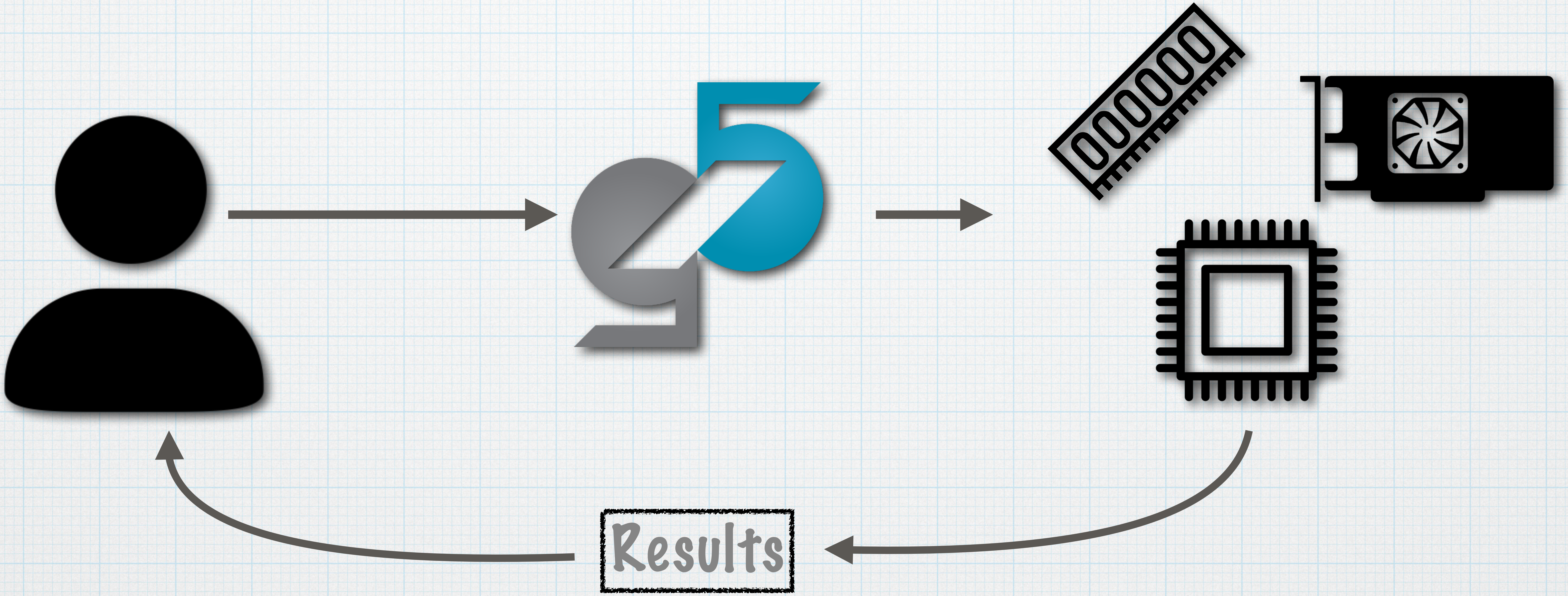
The Problem



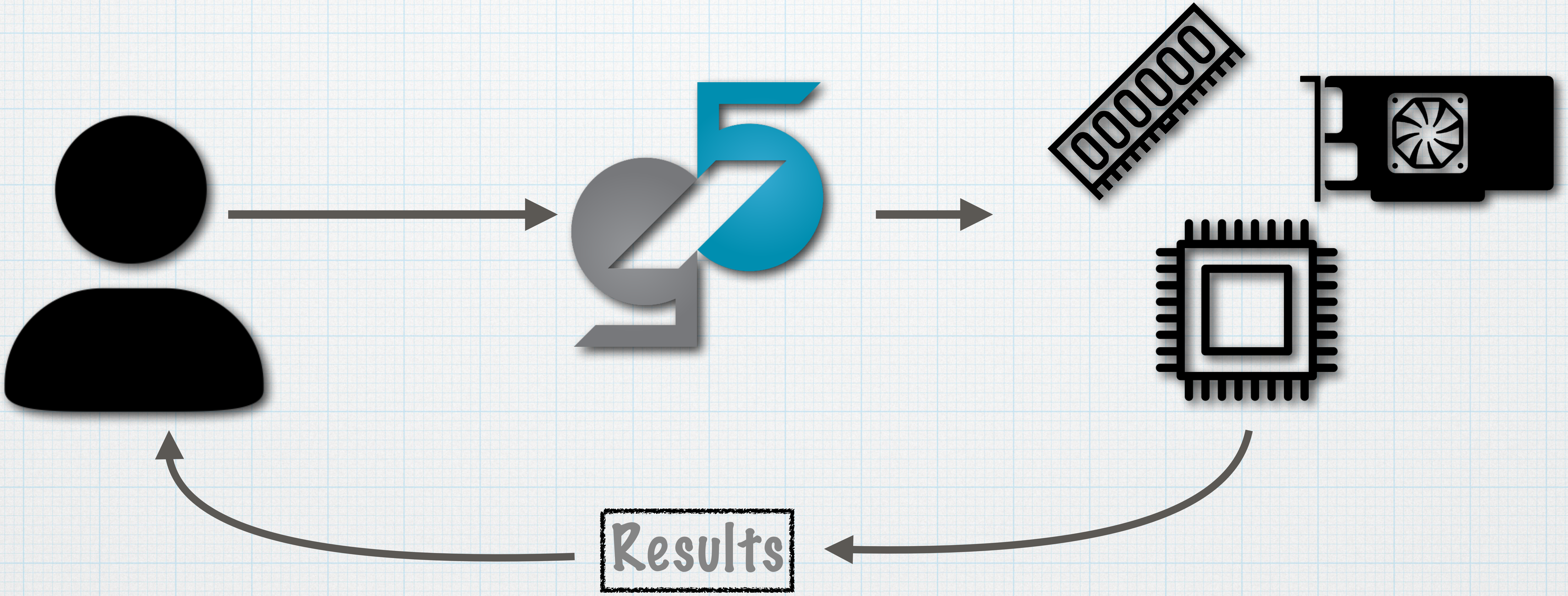
The Problem



The Problem

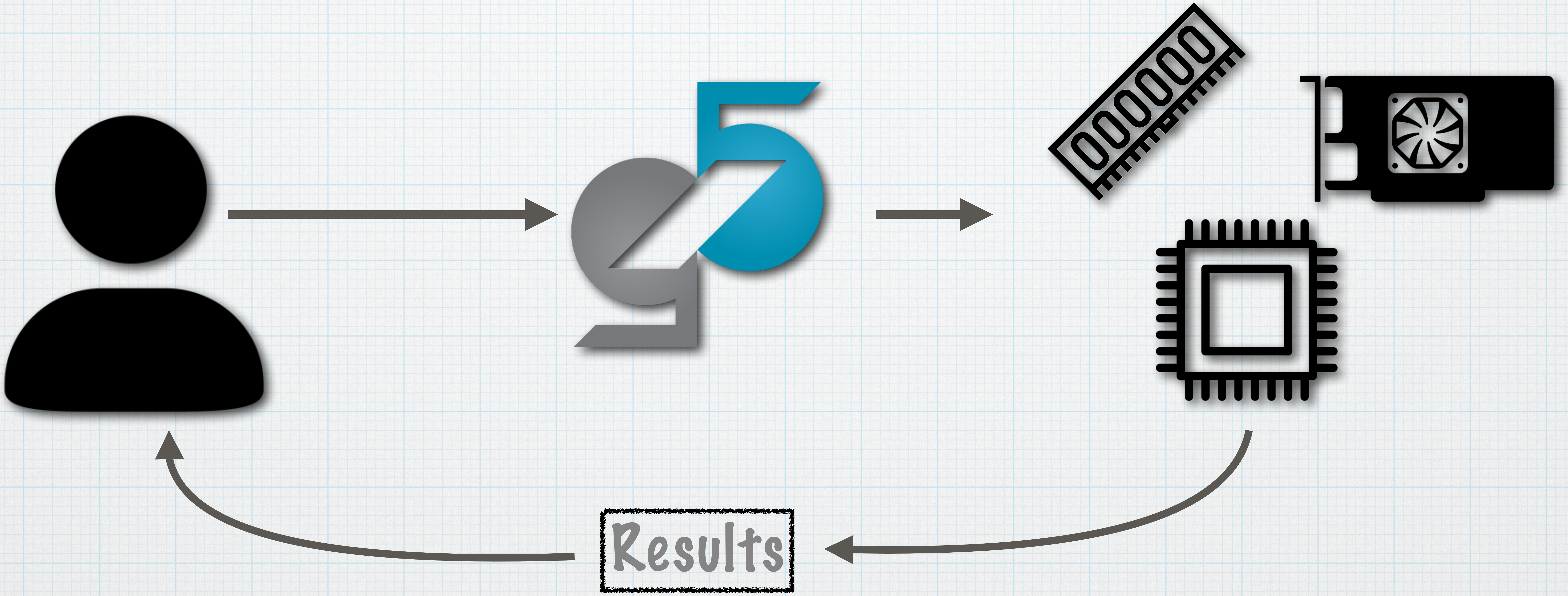


The Problem



How do we manage this information?

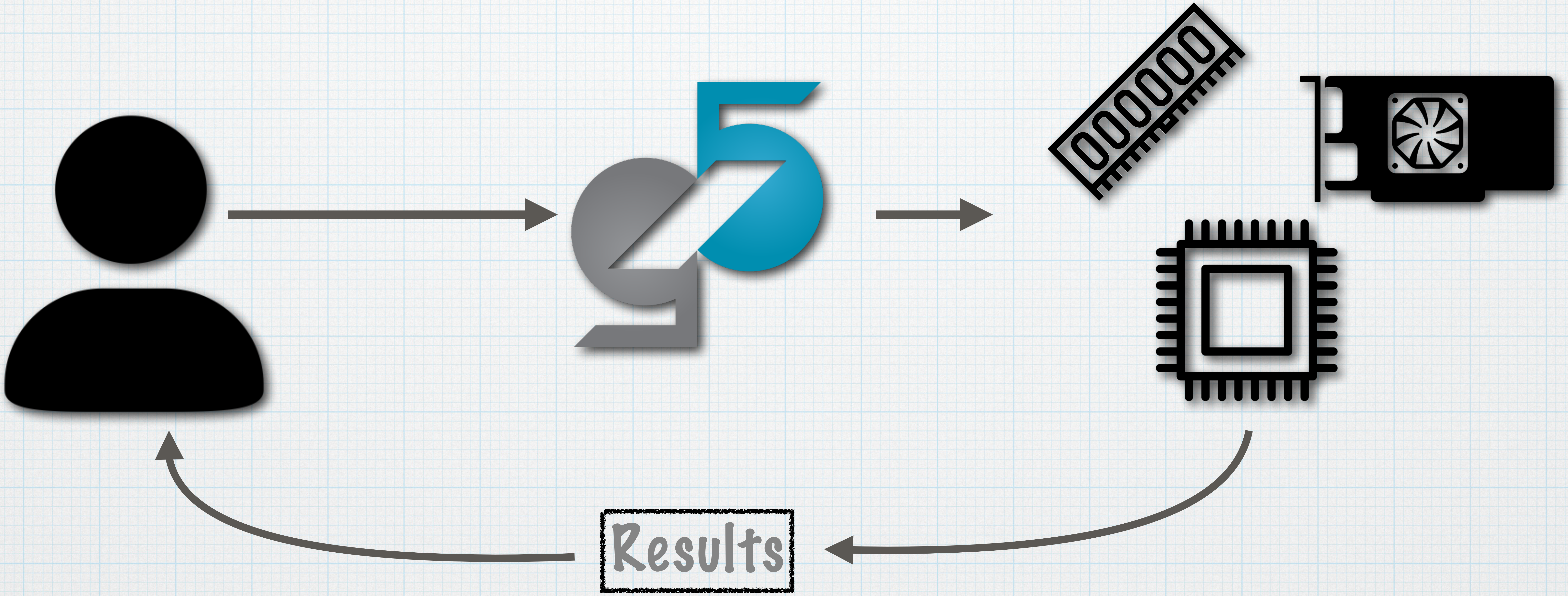
The Problem



How do we manage this information?

How do we easily reproduce what we've done?

The Problem



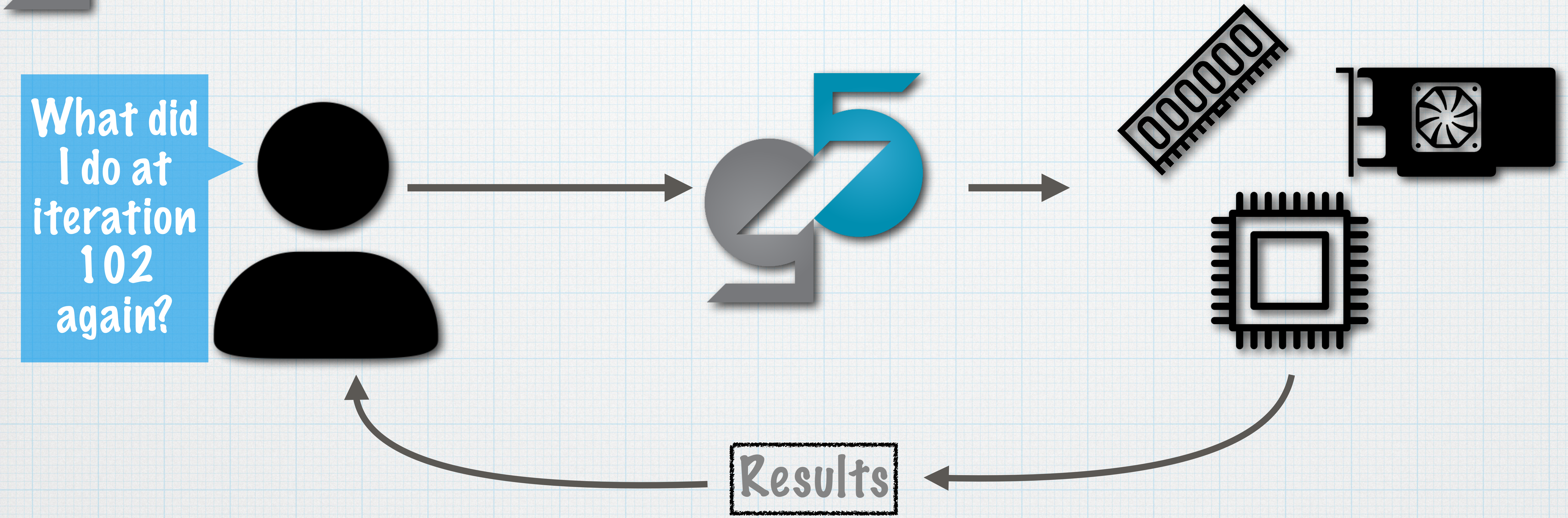
Results

How do we manage this information?

How do we easily reproduce what we've done?

How do we communicate our setup/results in a standardized way?

The Problem



What did I do at iteration 102 again?

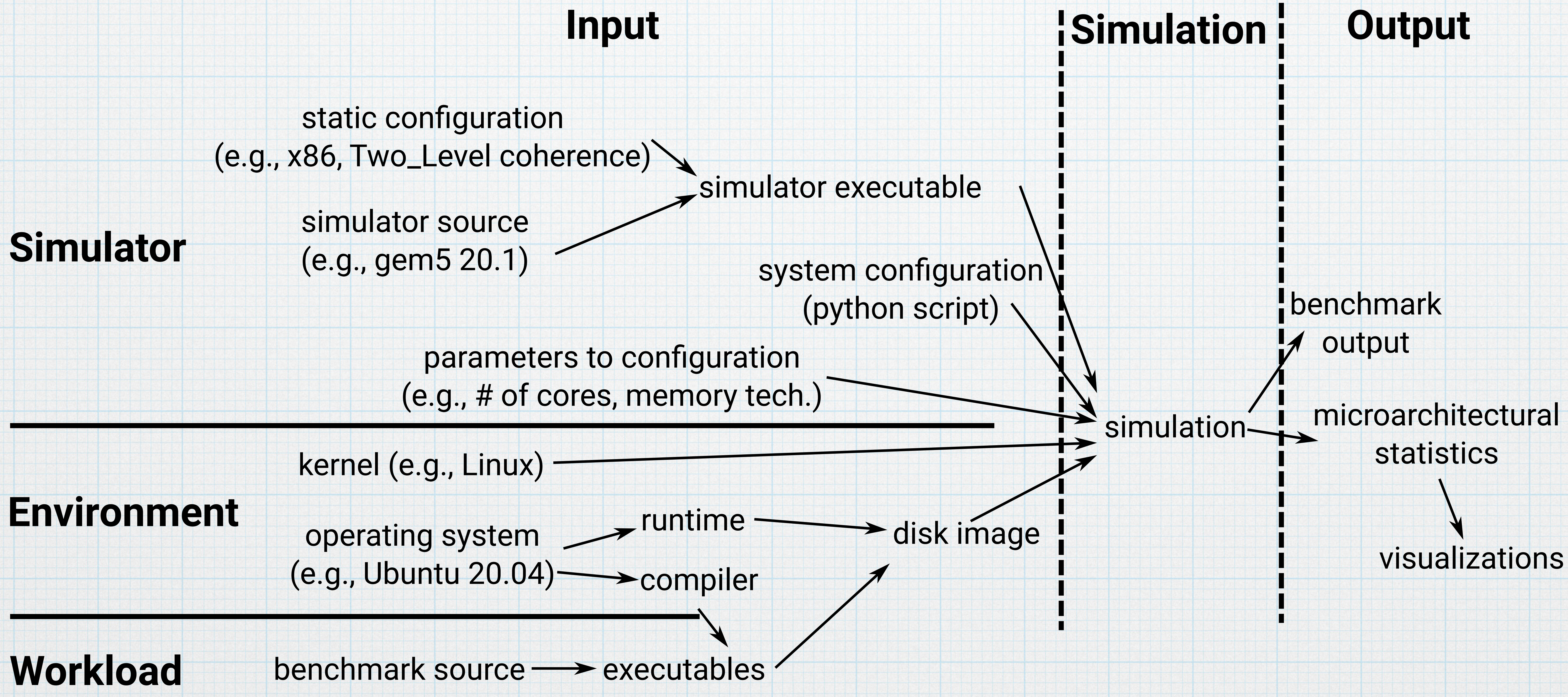
Results

How do we manage this information?

How do we easily reproduce what we've done?

How do we communicate our setup/results in a standardized way?

A Full-System workflow



For most experiments:

For most experiments:

Too many configurations!

For most experiments:

Too many configurations!

Too many results!

For most experiments:

Too many configurations!

Too many results!

No standardized way to communicate setups, or allow reproducibility.

For most experiments:

Too many configurations!

Too many results!

No standardized way to communicate setups, or allow reproducibility.

No official source of components/resources.

The Solution

The Solution

gem5art

- Artifacts.
- Reproducibility.
- Testing.

The Solution

gem5art

- Artifacts.
- Reproducibility.
- Testing.

The Solution

gem5art

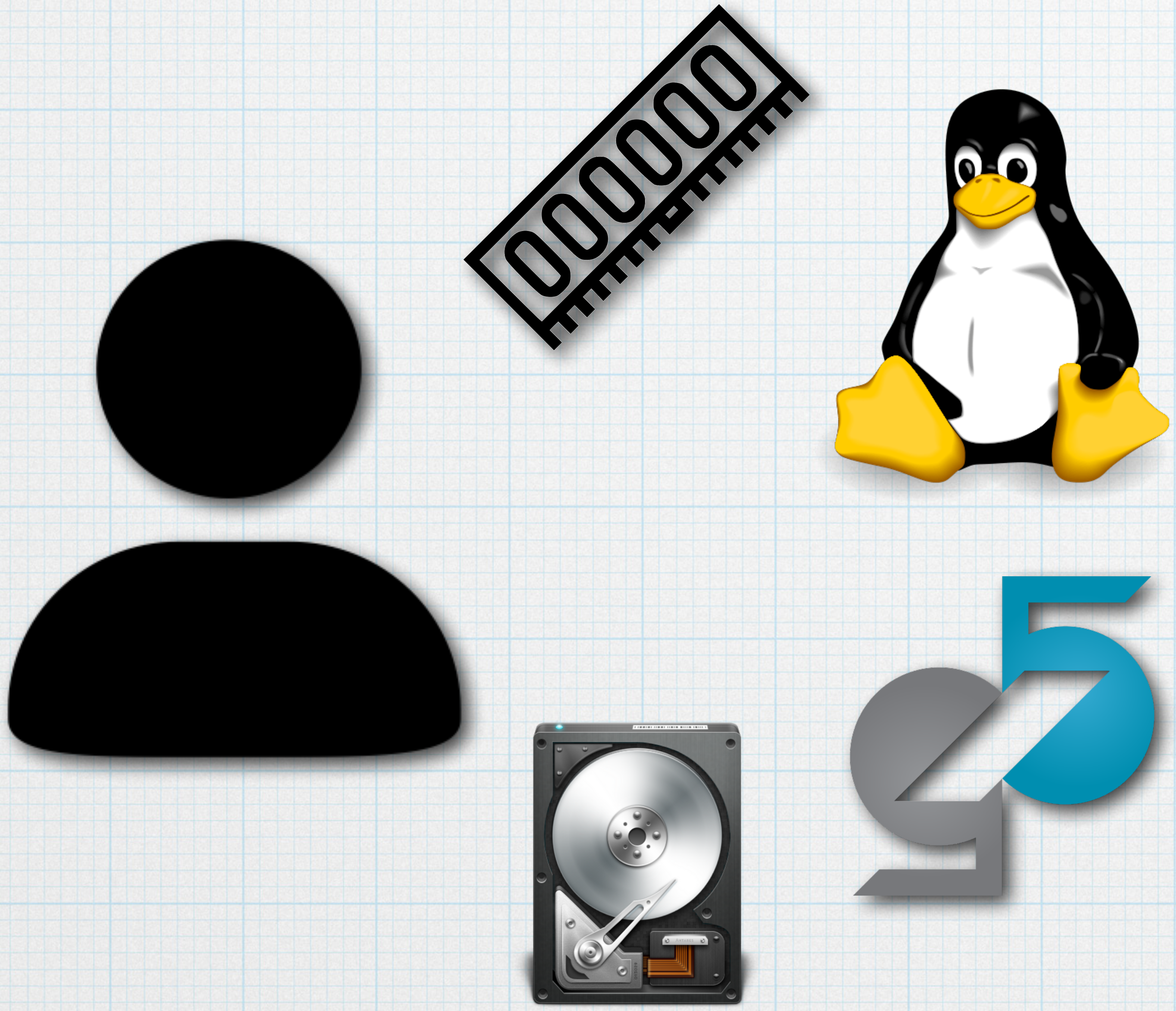
- Artifacts.
- Reproducibility.
- Testing.

gem5 Resources

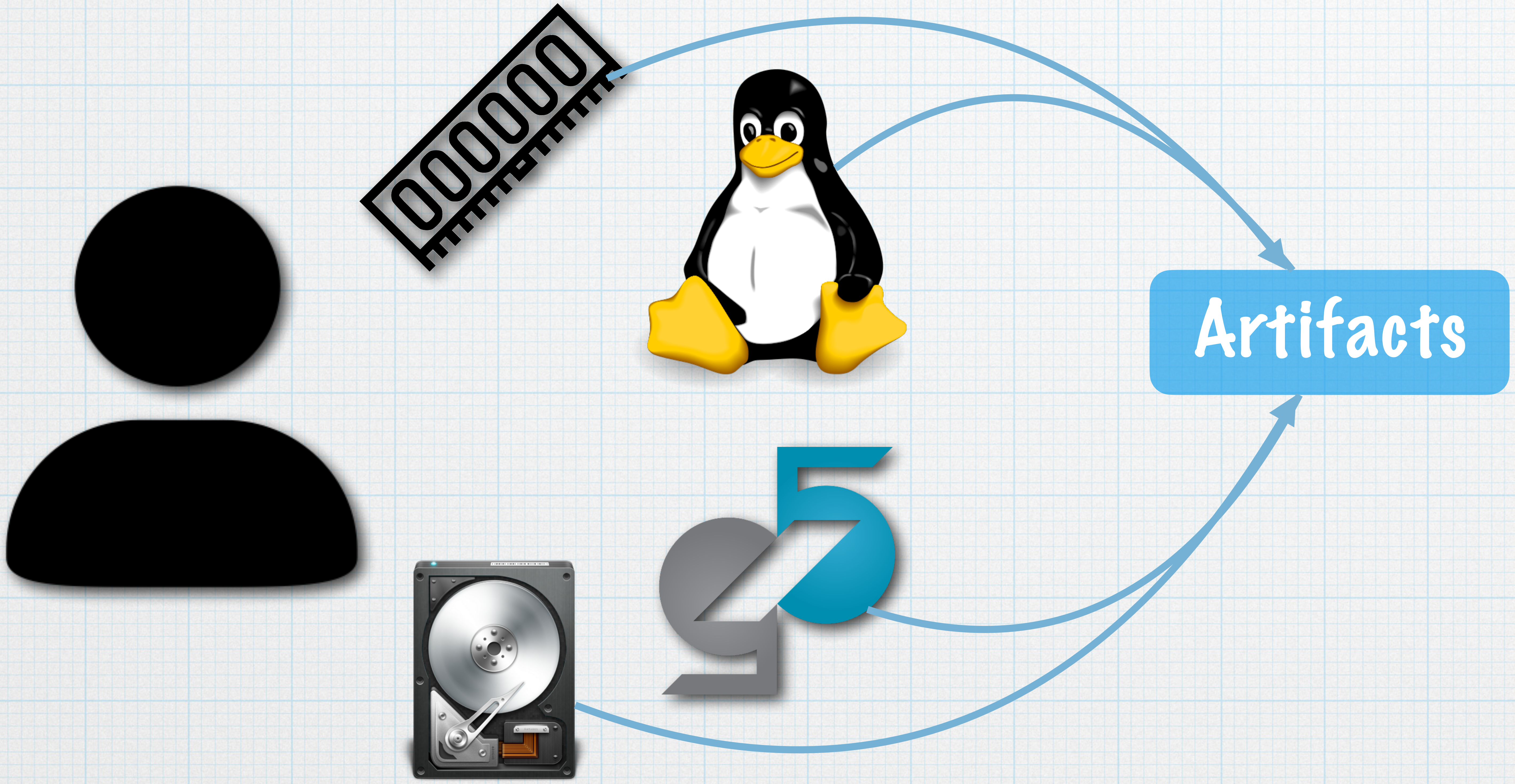
- Pre-built.
- gem5-compatible.
- Open-source, extendable.

gem5art:
A high-level overview

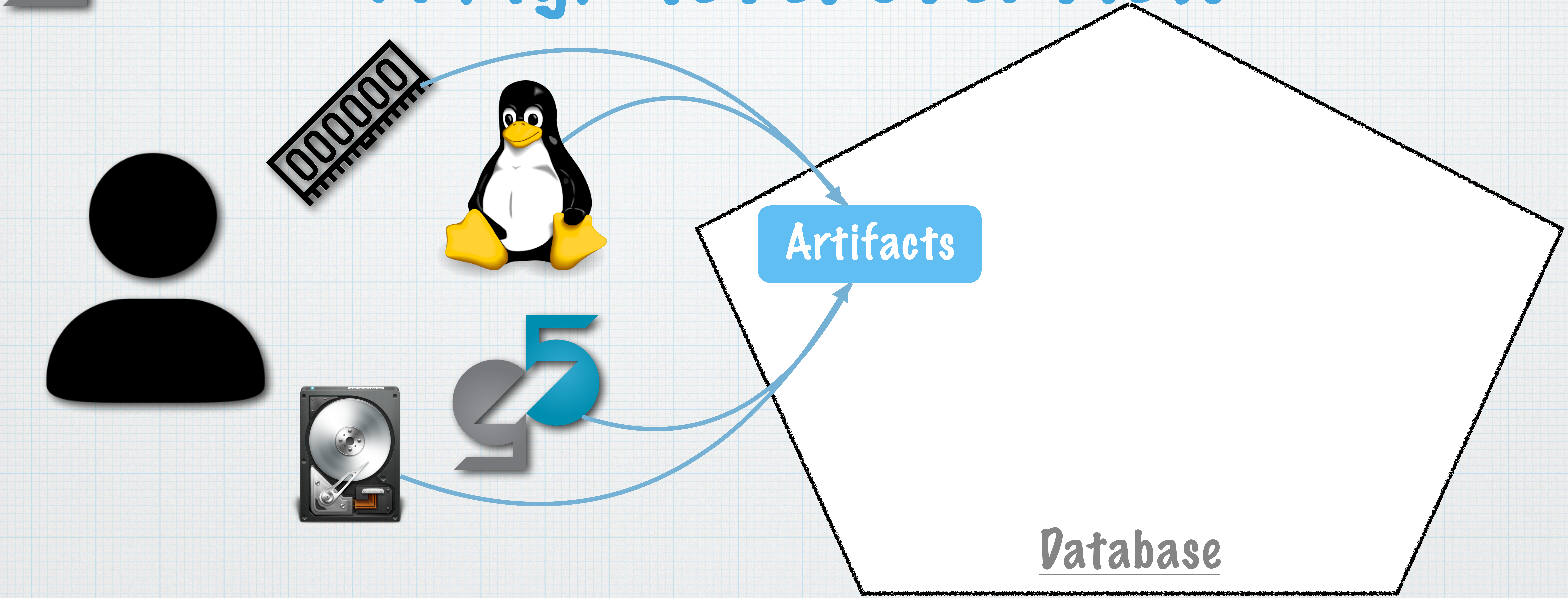
gem5art:
A high-level overview



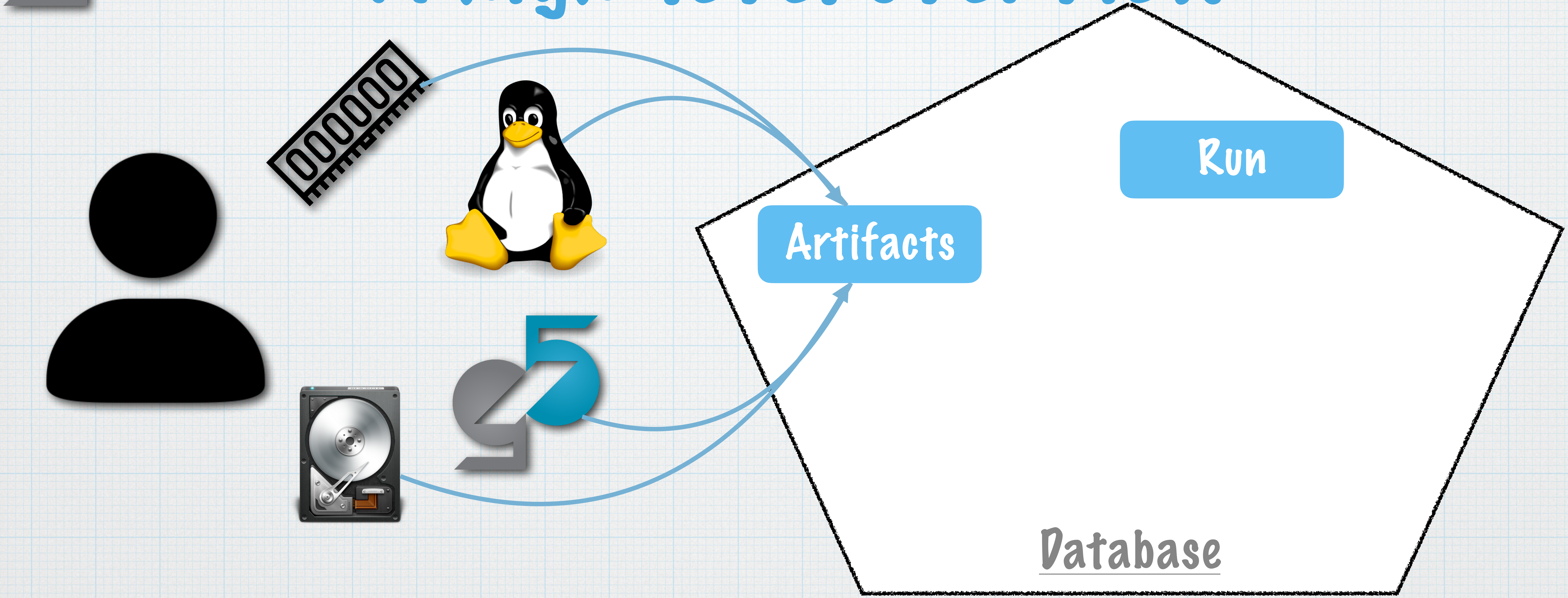
gem5art: A high-level overview



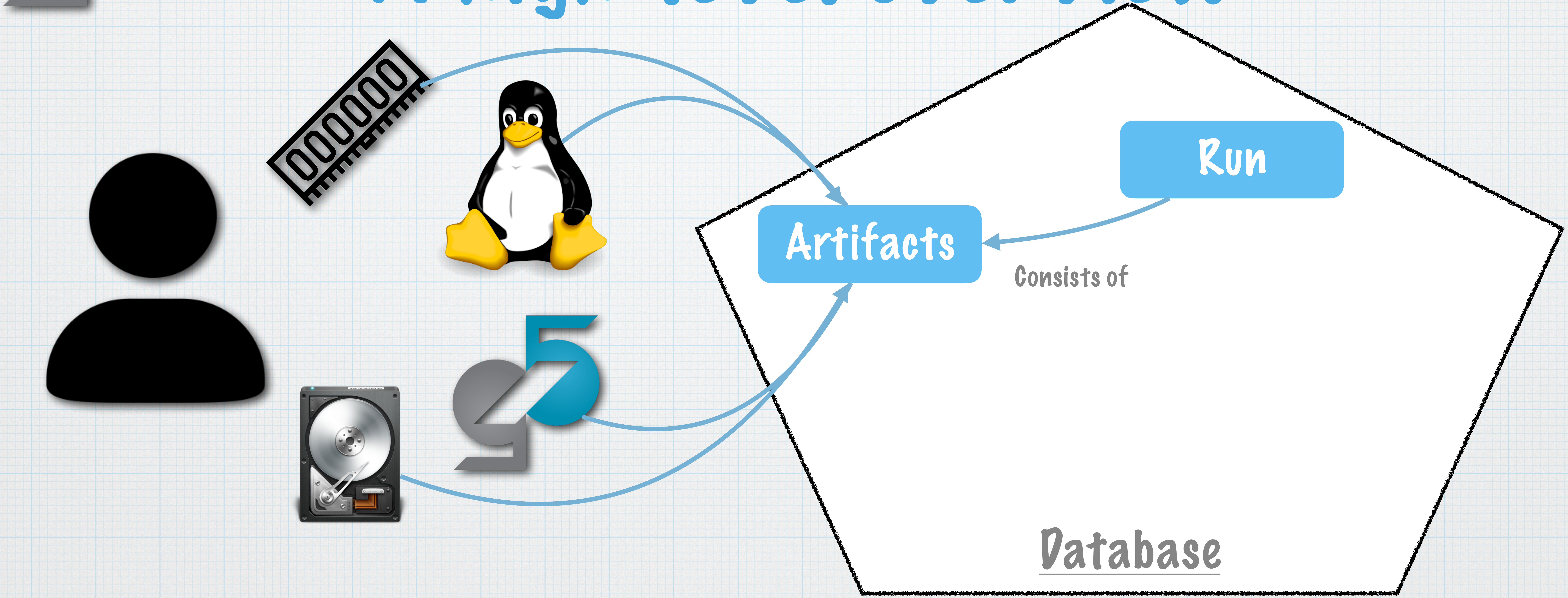
gem5art:
A high-level overview



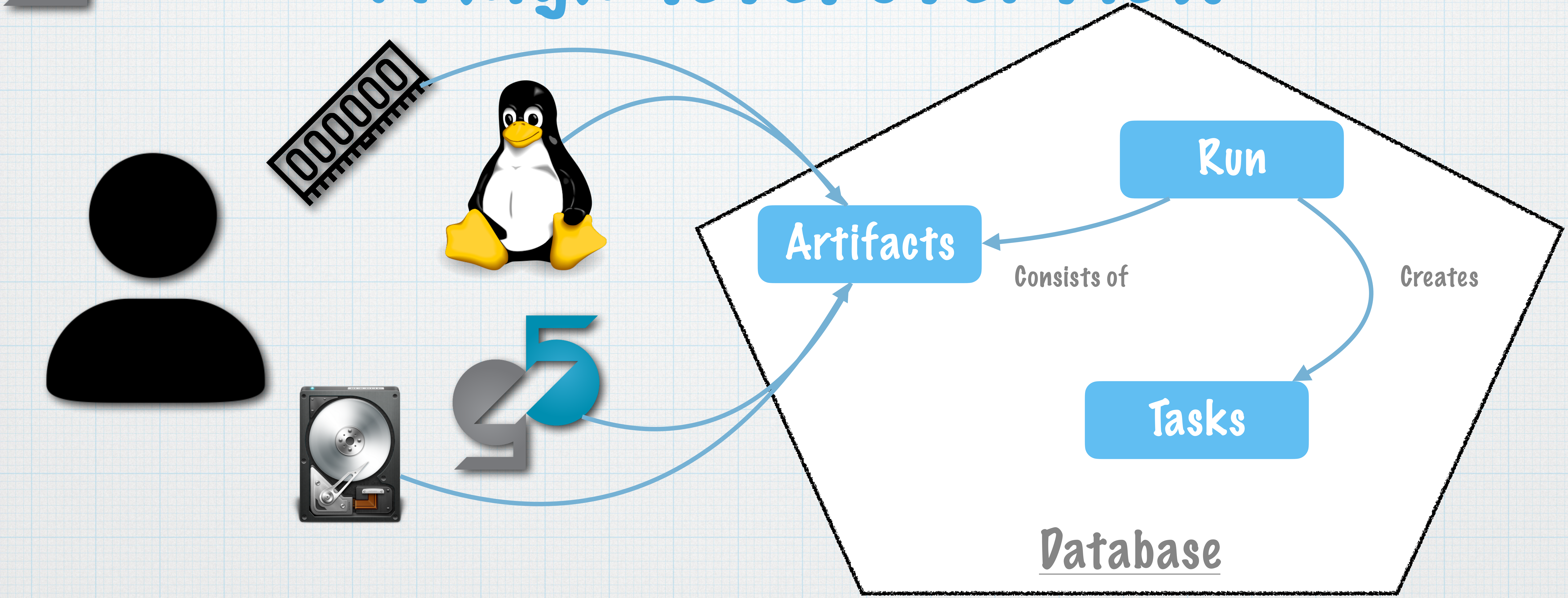
gem5art: A high-level overview



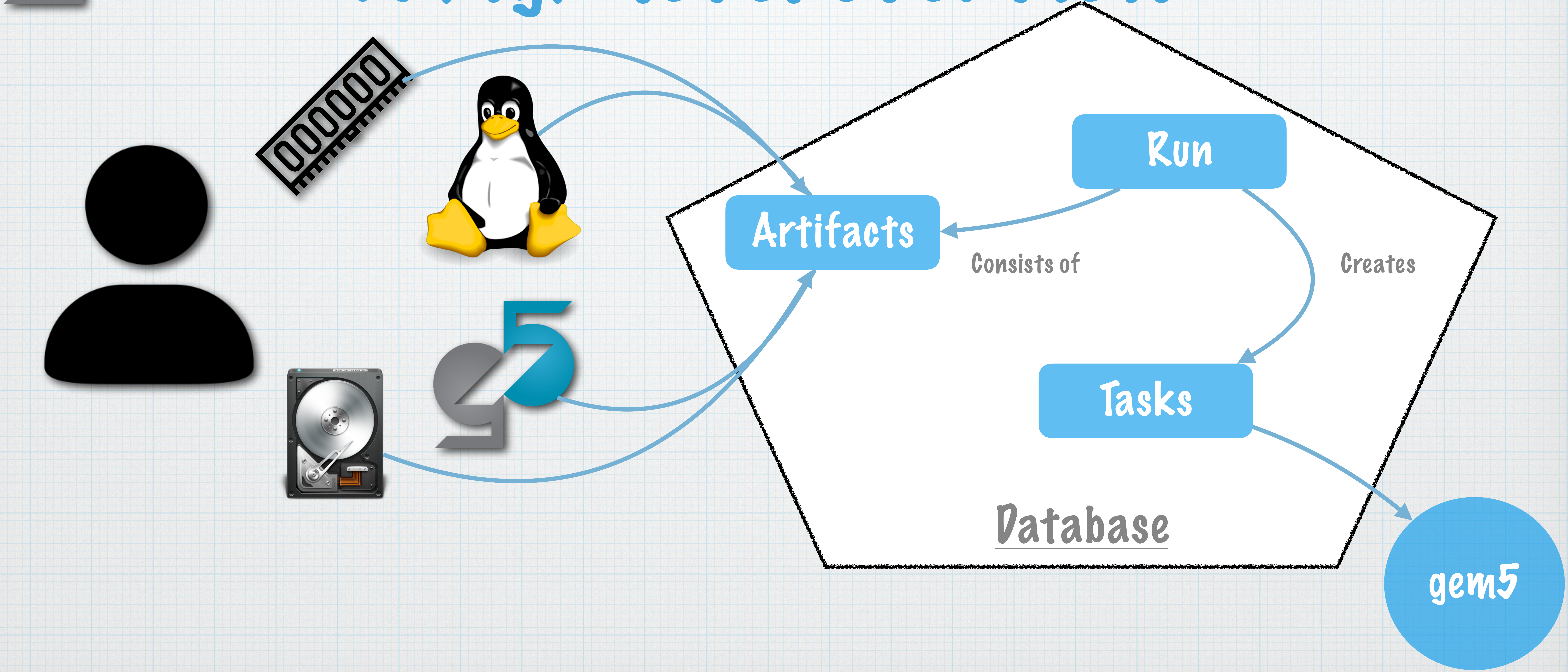
gem5art: A high-level overview



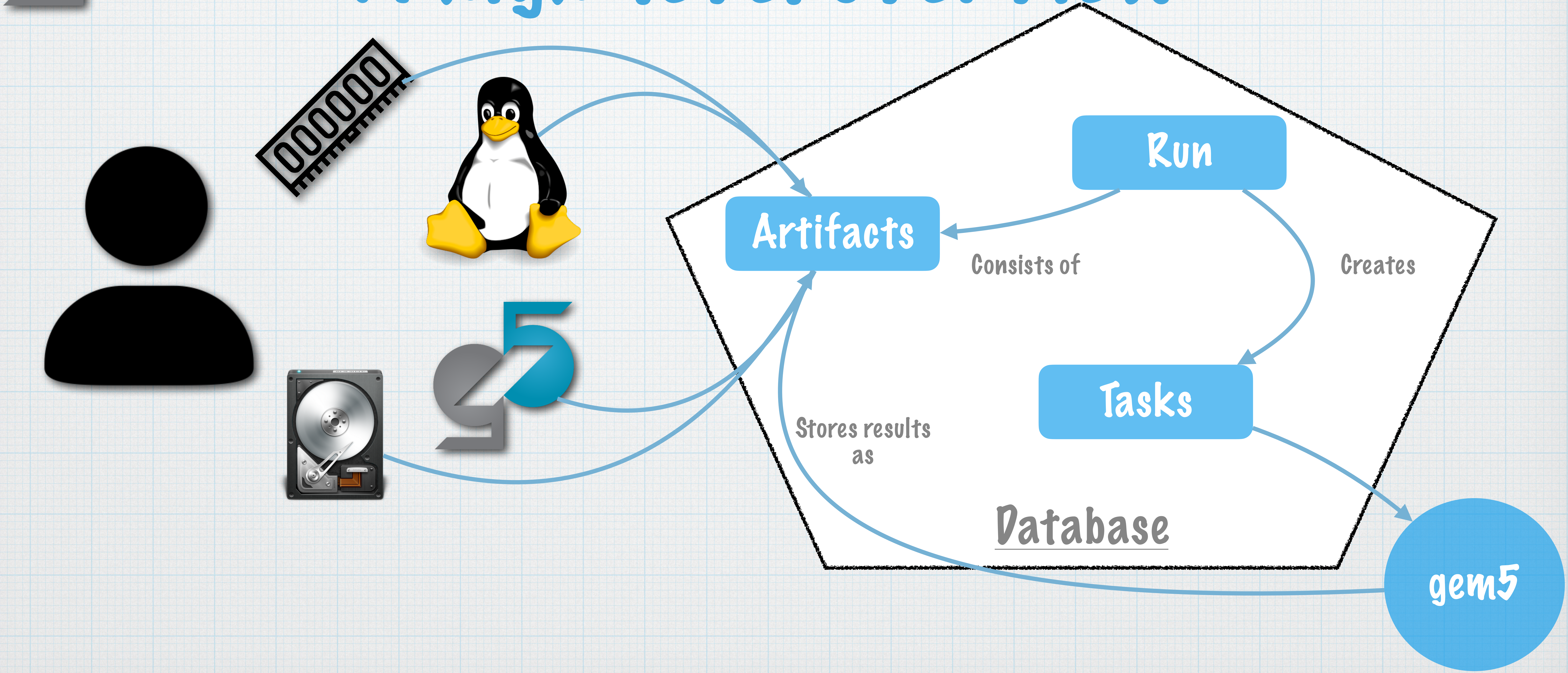
gem5art: A high-level overview



gem5art:
A high-level overview

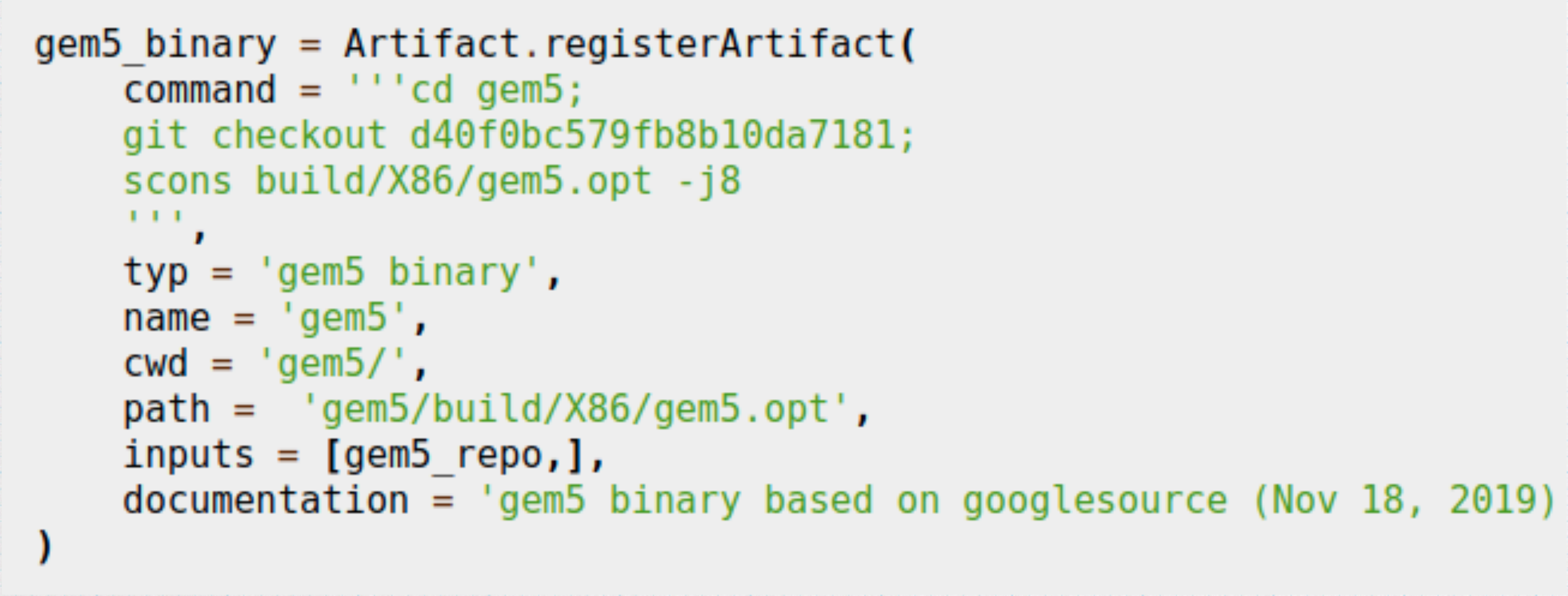


gem5art:
A high-level overview



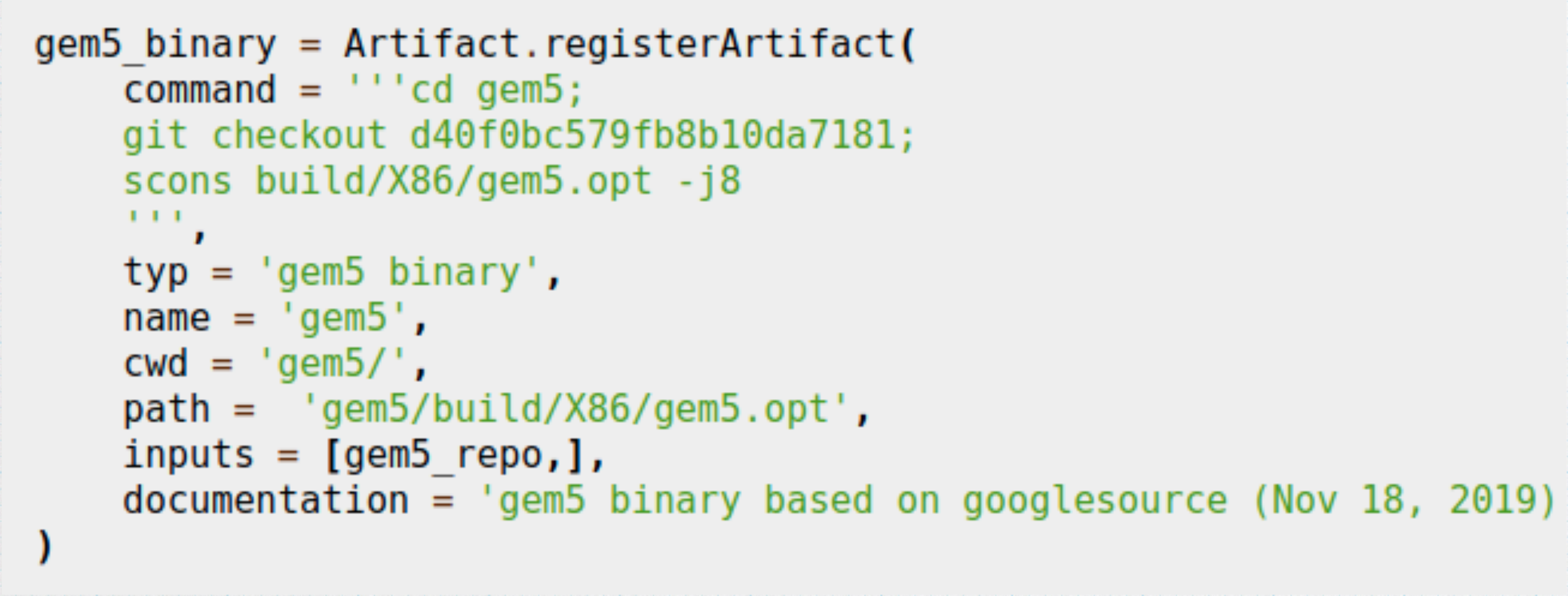
gem5art Workflow:
A slightly lower-level view

gem5art Workflow:
A slightly lower-level view

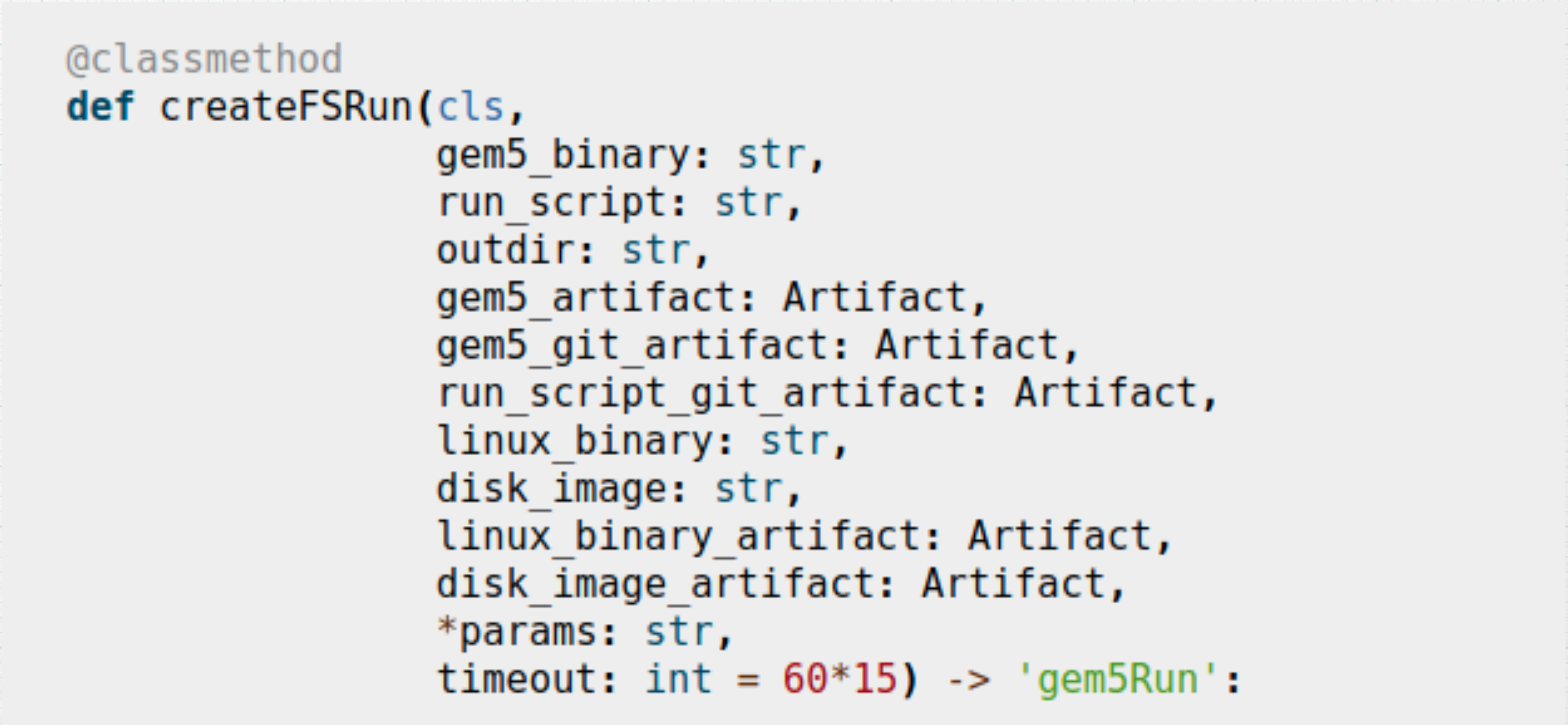


```
gem5_binary = Artifact.registerArtifact(
    command = '''cd gem5;
    git checkout d40f0bc579fb8b10da7181;
    scons build/X86/gem5.opt -j8
    ''',
    typ = 'gem5 binary',
    name = 'gem5',
    cwd = 'gem5/',
    path =  'gem5/build/X86/gem5.opt',
    inputs = [gem5_repo,],
    documentation = 'gem5 binary based on googlesource (Nov 18, 2019)
)
```

gem5art Workflow:
A slightly lower-level view

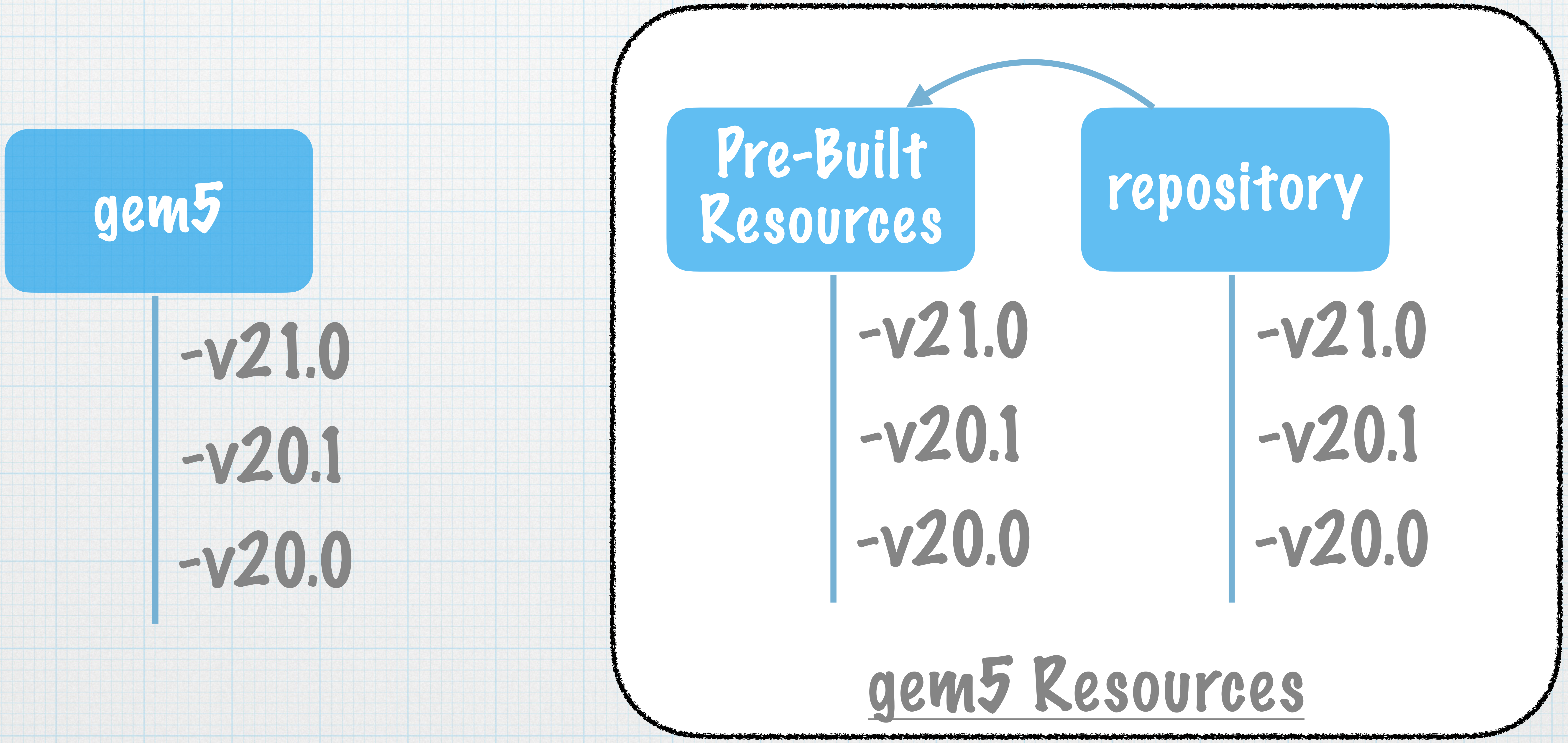


```
gem5_binary = Artifact.registerArtifact(
    command = '''cd gem5;
    git checkout d40f0bc579fb8b10da7181;
    scons build/X86/gem5.opt -j8
    ''',
    typ = 'gem5 binary',
    name = 'gem5',
    cwd = 'gem5/',
    path =  'gem5/build/X86/gem5.opt',
    inputs = [gem5_repo,],
    documentation = 'gem5 binary based on googlesource (Nov 18, 2019)
)
```



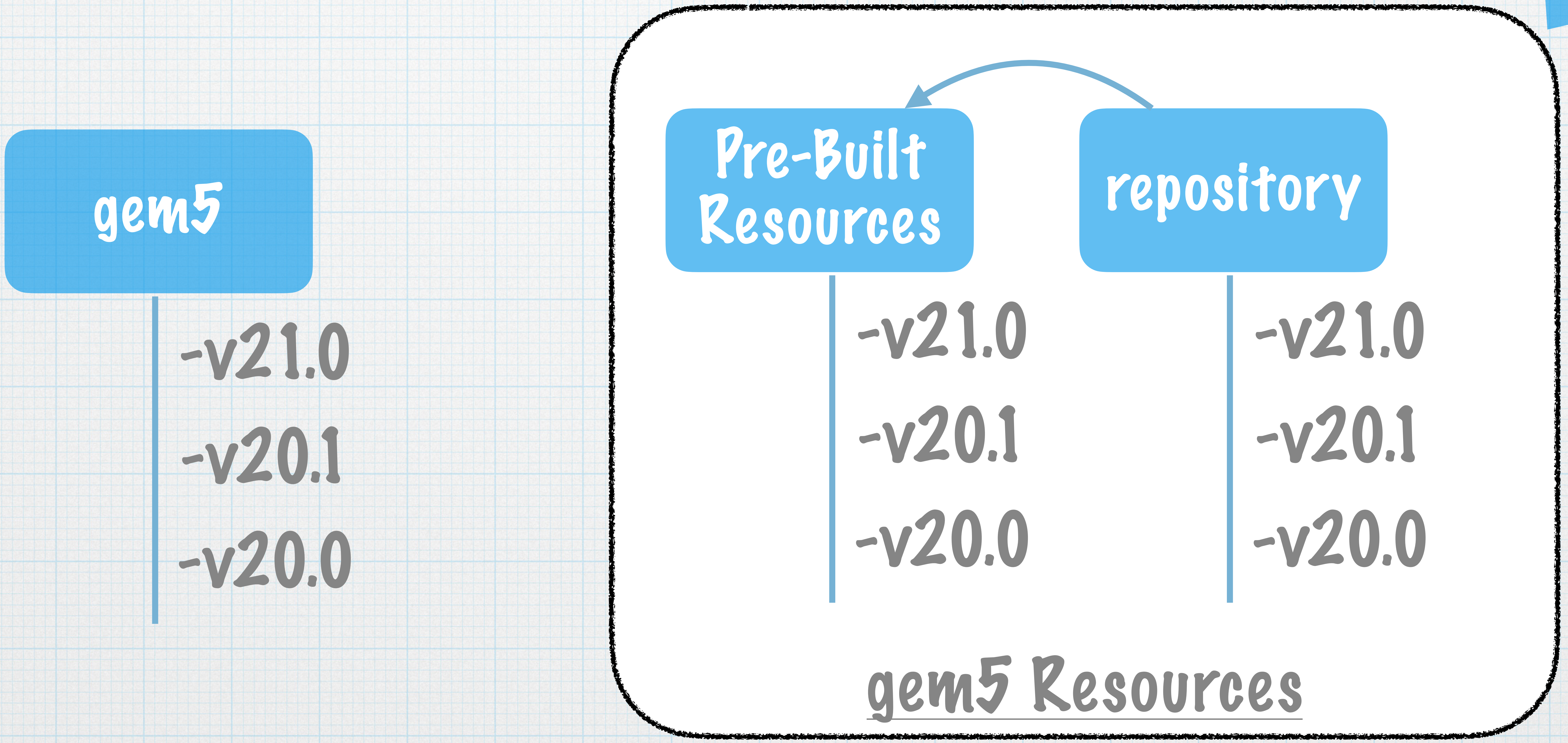
```
@classmethod
def createFSRun(cls,
                gem5_binary: str,
                run_script: str,
                outdir: str,
                gem5_artifact: Artifact,
                gem5_git_artifact: Artifact,
                run_script_git_artifact: Artifact,
                linux_binary: str,
                disk_image: str,
                linux_binary_artifact: Artifact,
                disk_image_artifact: Artifact,
                *params: str,
                timeout: int = 60*15) -> 'gem5Run':
```

gem5 Resources: A high-level overview



gem5 Resources:
A high-level overview

16+ apps/ benchmarks/ images



gem5
- -v21.0
- -v20.1
- -v20.0

Pre-Built Resources
- -v21.0
- -v20.1
- -v20.0

repository
- -v21.0
- -v20.1
- -v20.0

gem5 Resources

Use-Case 1: Co-Design

Use-Case 1: Co-Design

"How does the execution time of PARSEC applications change between Ubuntu 18.04 and 20.04, for single core and 8 core CPU setups?"

Use-Case 1: Co-Design

Use-Case 1: Co-Design

Moving parts:

Operating System: Ubuntu 18.04, Ubuntu 20.04

Applications: 10 benchmark applications

Num Processors: Single Core, 8 Core

Use-Case 1: Co-Design

Moving parts:

Operating System: Ubuntu 18.04, Ubuntu 20.04

Applications: 10 benchmark applications

Num Processors: Single Core, 8 Core

This produces a total of 40 runs.

Use-Case 1: Co-Design

Moving parts:

Operating System: Ubuntu 18.04, Ubuntu 20.04
Applications: 10 benchmark applications
Num Processors: Single Core, 8 Core

This produces a total of 40 runs.

Each run produces results. In this case we concern ourselves with execution time.

Use-Case 1: Co-Design

Use-Case 1: Co-Design

With gem5art/gem5 resources this was easy

Use-Case 1: Co-Design

With gem5art/gem5 resources this was easy

1) Obtain the Parsec Benchmark from gem5 resources

Use-Case 1: Co-Design

With gem5art/gem5 resources this was easy

1) Obtain the Parsec Benchmark from gem5 resources

2) Register artifacts

Use-Case 1: Co-Design

With gem5art/gem5 resources this was easy

1) Obtain the Parsec Benchmark from gem5 resources
2) Register artifacts
3) Create a Run Script

Use-Case 1: Co-Design

With gem5art/gem5 resources this was easy

1) Obtain the Parsec Benchmark from gem5 resources

2) Register artifacts

3) Create a Run Script

4) Execute

Use-Case 1: Co-Design

With gem5art/gem5 resources this was easy

1) Obtain the Parsec Benchmark from gem5 resources

2) Register artifacts

3) Create a Run Script

4) Execute

5) Query the database for desired results

Use-Case 1: Co-Design

With gem5art/gem5 resources this was easy

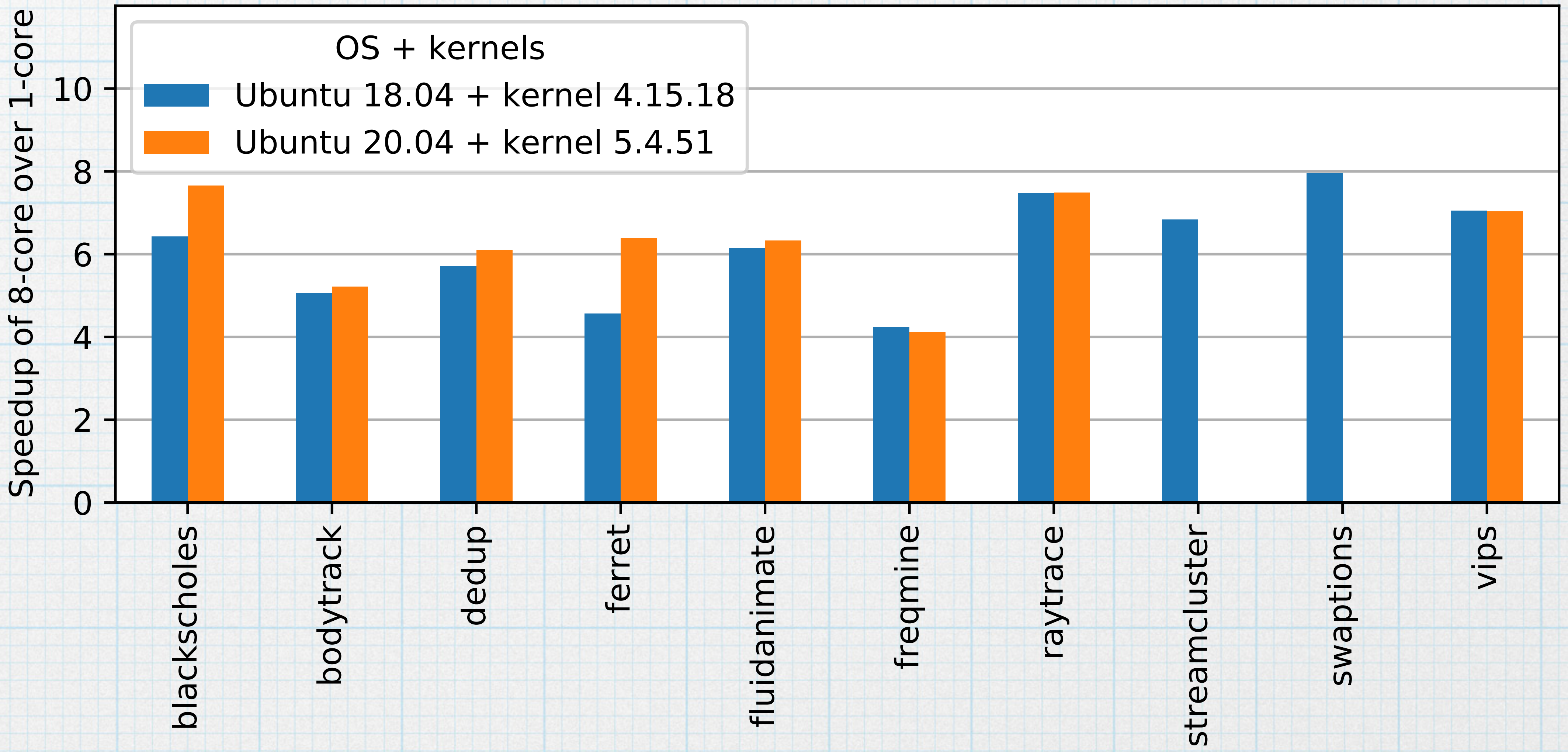
1) Obtain the Parsec Benchmark from gem5 resources

2) Register artifacts

3) Create a Run Script

4) Execute

5) Query the database for desired results



Use-Case 2: Testing!

Use-Case 2: Testing!

"How does gem5 perform when booting Linux on different architecture setups?"

Use-Case 2: Testing!

"How does gem5 perform when booting Linux on different architecture setups?"

This is a common gem5 test

Use-Case 2: Testing

Use-Case 2: Testing

Moving parts:

Kernel: 4.4.186, 4.9.186, 4.14.134, 4.19.84, 5.4.49

Num Processors: 1, 2, 4, 8

CPU Models: kvm, atomic, simple, o3

Memory System: classic, MI_Example, MESI_Two_Level

Boot: Kernel Only, Full Ubuntu

Use-Case 2: Testing

Moving parts:
Kernel: 4.4.186, 4.9.186, 4.14.134, 4.19.84, 5.4.49
Num Processors: 1, 2, 4, 8
CPU Models: kvm, atomic, simple, o3
Memory System: classic, MI_Example, MESI_Two_Level
Boot: Kernel Only, Full Ubuntu

This produces a total of 480 runs.

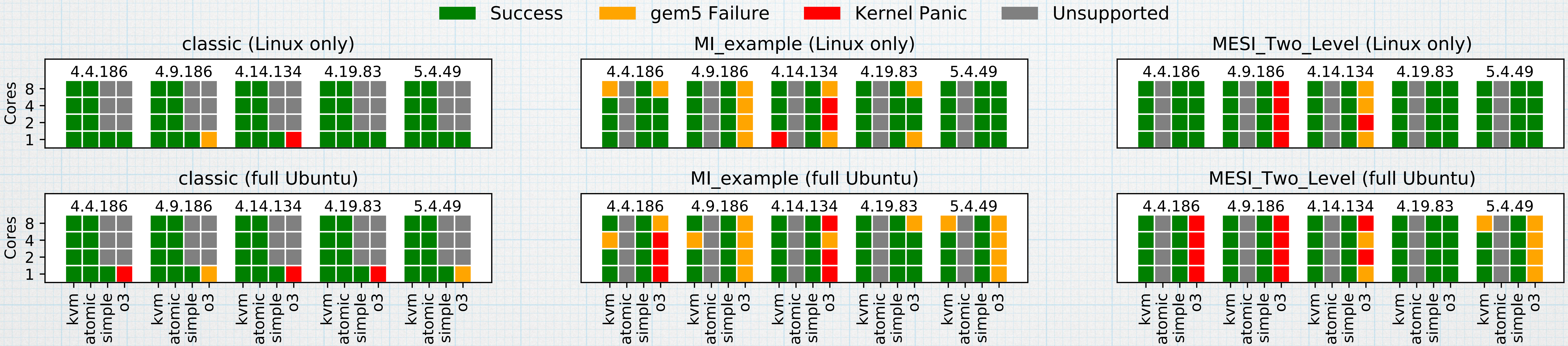
Use-Case 2: Testing

Moving parts:
Kernel: 4.4.186, 4.9.186, 4.14.134, 4.19.84, 5.4.49
Num Processors: 1, 2, 4, 8
CPU Models: kvm, atomic, simple, o3
Memory System: classic, MI_Example, MESI_Two_Level
Boot: Kernel Only, Full Ubuntu

This produces a total of 480 runs.

For each run we wish to keep track of whether the run was a success, there was a gem5 error, or a kernel panic.

Use-Case 2: Testing



Thank you!

Enabling Reproducible and Agile Full-System Simulation

Work by Bobby R. Bruce, Ayaz Akram, Hoa Nguyen, Kyle Roarty, Mahyar Samani, Marjan Fariborz, Trivikram Reddy, Matthew D. Sinclair, and Jason Lowe-Power

Artifact: https://doi.org/10.6084/m9.figshare.14176802

Paper at: https://arch.cs.ucdavis.edu/assets/papers/ispass21-gem5art.pdf

Research supported by: NSF Grants CNS-1925724 and CNS-1850566

THE UNIVERSITY OF CALIFORNIA · DAVIS · LET THERE BE LIGHT

WISCONSIN
UNIVERSITY OF WISCONSIN–MADISON

AMD

Presented by Bobby R. Bruce