Turning Models Into Applications– GAMS Engine and GAMS Transfer

Steve Dirkse

Adam Christensen

GAMS Engine

Solving Models On Centralized Compute Resources

Steve Dirkse   Frederik Proske   Hamdi Burak Usul

GAMS Company

Staff

- 25 full time + students (US & EU)
- PhD level modeling and optimization experts with startup mentality
- Experienced management, generational change in 2015
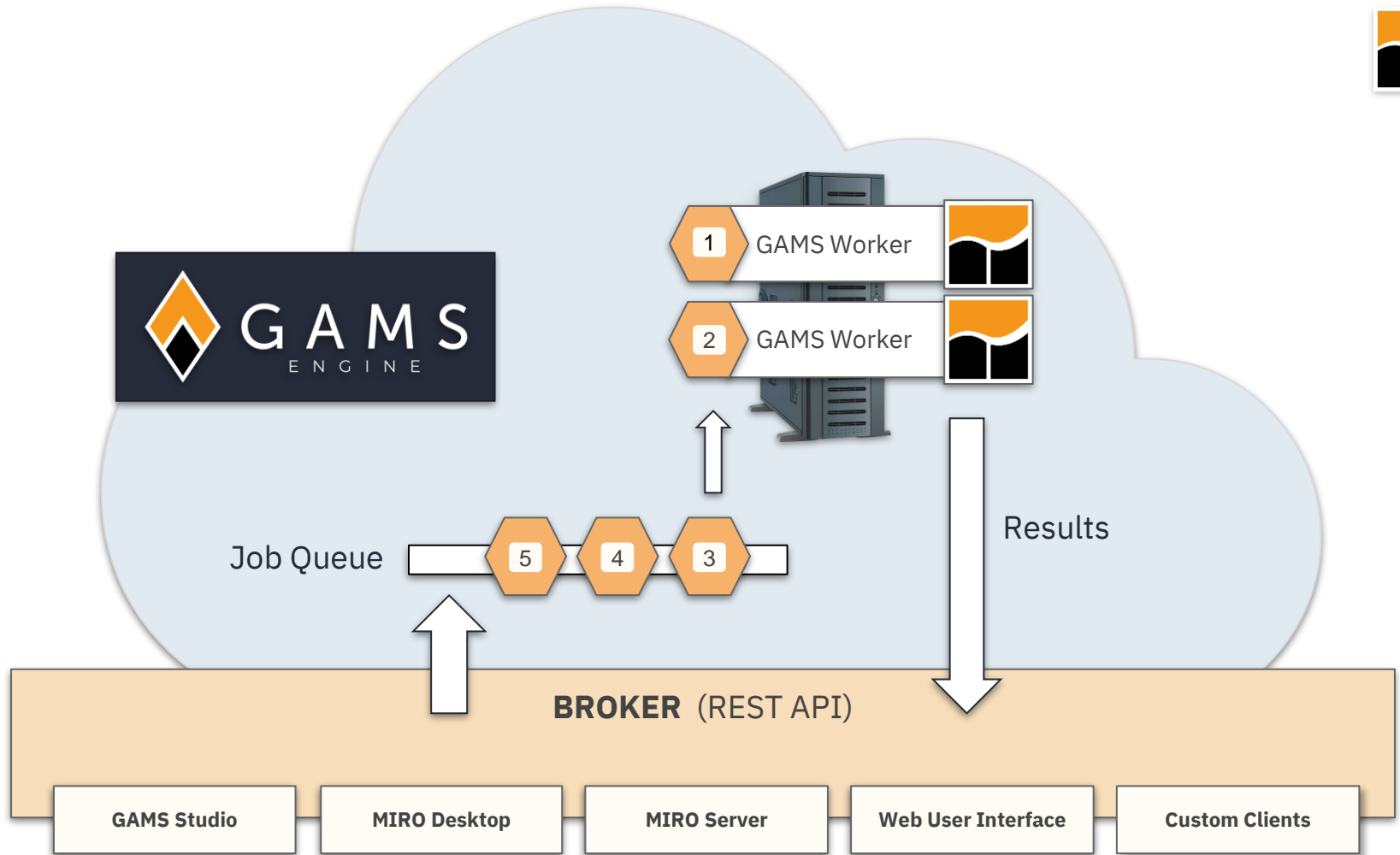

Competencies & Activities

- First algebraic modeling software
- Platform independent software development
- Software quality assurance
- Support for software and bundled solver products
- Consulting services
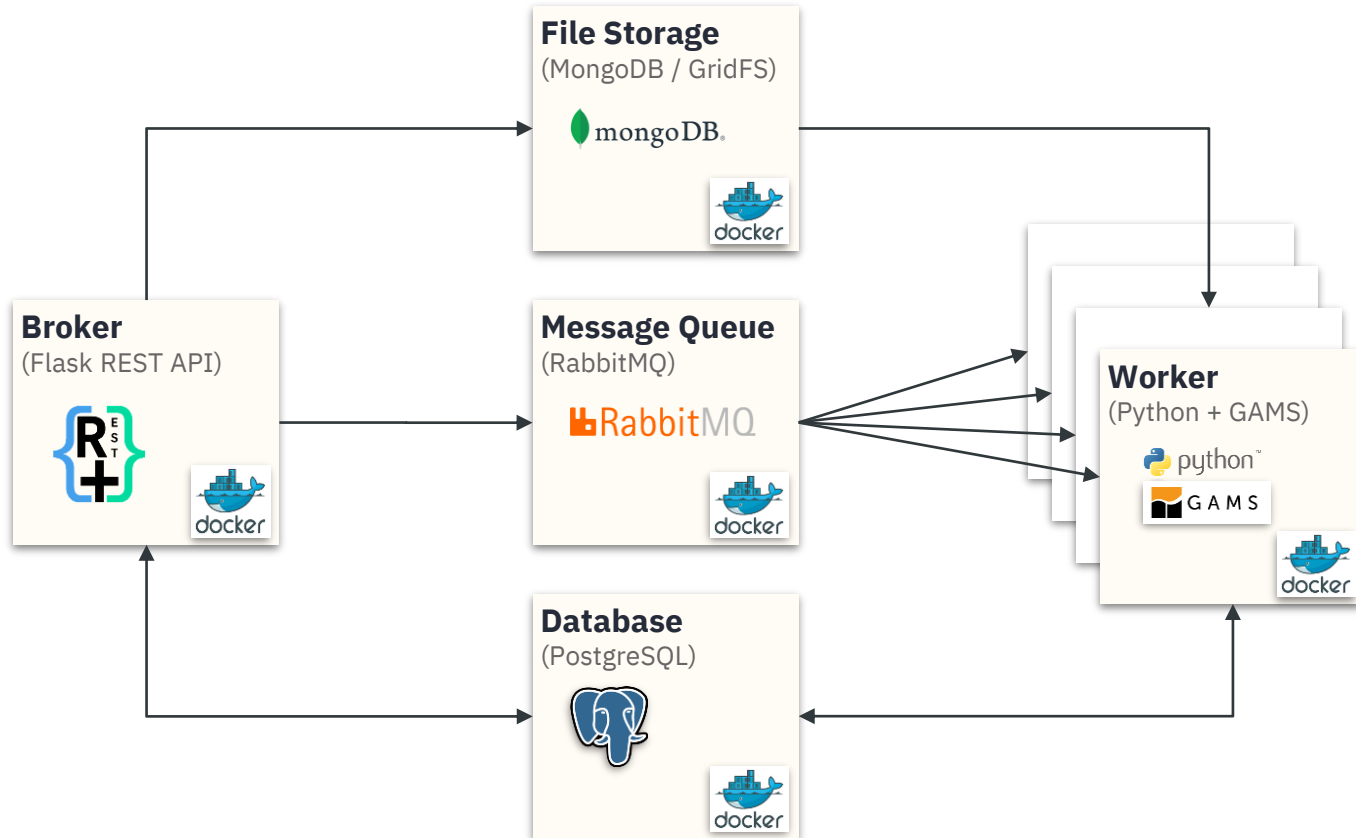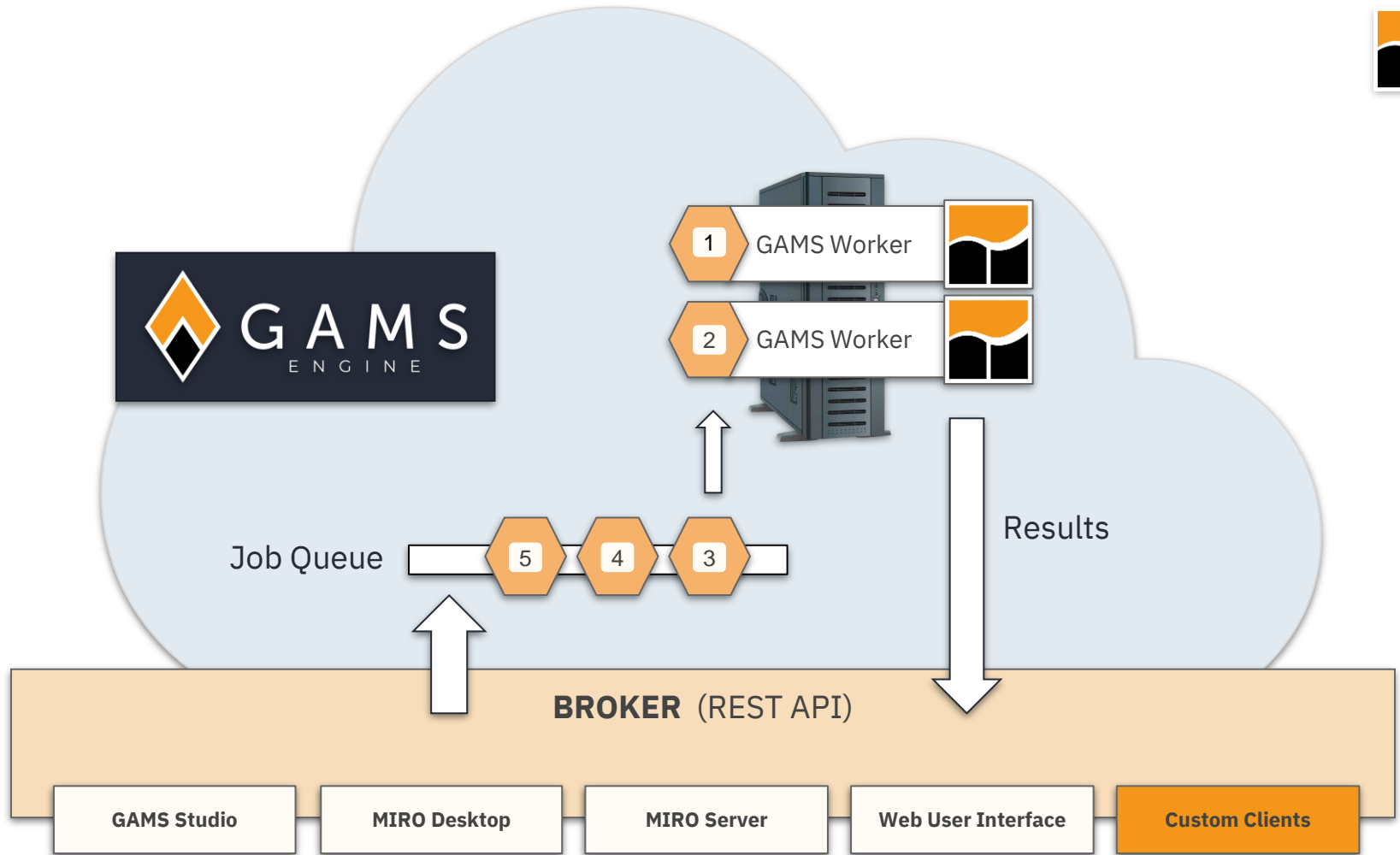- Projects with extensive partner network

Customers

- Global user community
- Academic and commercial
- Energy, economics, logistics, operations research

GAMS Engine Docker stack

File Storage
(MongoDB / GridFS)

Broker
(Flask REST API)

Message Queue
(RabbitMQ)

Worker
(Python + GAMS)

Database
(PostgreSQL)

GAMS ENGINE

1 GAMS Worker

2 GAMS Worker

Job Queue 5 4 3

Results

BROKER (REST API)

| GAMS Studio | MIRO Desktop | MIRO Server | Web User Interface | **Custom Clients** |

Python demo - custom clients

- Prerequisites for this demo
    - An updated Windows machine with WSL (Linux also works)
    - docker and docker-compose installed
    - Python3
- Demo steps
    - Engine install (review only)
    - Engine start
    - Engine licensing (Web UI)
    - Job start, namespace, job register (Web UI)
    - API generation (review only)
    - Python - listing jobs
    - Python - create a job, wait, retrieve results, delete results from Engine

Benefits

- Use of powerful server hardware
- Job scheduling is built in (4 workers in our example case)
- Clients can do other things or go offline and fetch results later
- Centralised license management (especially useful in commercial settings)

Acknowledgements

Hamdi Burak Usul

Frederik Proske

Robin Schuchmann