# Warning - Opinions Ahead!



# Our goals for today

To **share** findings and insights from our work To **get** reactions, insights, questions and ideas

# Entry poll

a) Have you ever worked in a project where deliveries have been difficult or even impossible to test as a whole?

b) Have you ever ended up being **short of time** to complete the planned / needed pre-deployment testing?

c) Have you ever deployed code that you know is incompletely tested in one or more aspects\*?

\* features, correctness, performance, scalability, resilience, ...

# The Next Big Thing

# Full-scale, high-fidelity simulation

# **Full-scale**

Run at volumes and load levels that are at or above the highest observed real world load.

Every code or configuration change may invalidate previously executed tests

A system is essentially untested for any load level above what has been actually tested

# **High-fidelity**

# Provide interfaces and behavior with a level of detail that is indistinguishable\* from the real world system

\*Compare with modern games and the resolution of the eye

# Real or simulated?



### The Matrix



# **Our Vision**

# To provide **significant quality improvements** in transaction systems using simulation as a service

# Why is simulation needed?

Today's transaction systems are **highly interconnected** and have **complex architectures** and **large feature sets**.

Current test tools and methodologies don't solve the problem.

Simulation of production environments is required to trigger **fatal errors due to unknown flaws** before deployment to production.

# Unit testing

#### Useful to establish component level quality Limited scope and poor scalability

It will only take you so far ...

## Integration testing

#### Useful to verify connectivity Large overhead and poor scalability

It will only take you so far ...

# Our challenge

Designing from first principles Avoiding non-problems Representative synthetic scenarios Scalability

### Feature subsets

Simulations need not capture all of the real world system

Each simulation scenario must however be as similar as possible in all relevant aspects

Most of the features can be implemented with limited resources

The last features will require much more resources

# Scalability

RAM, disk, network bandwidth, CPU cycles
Self provisioning, auto-configuration
Slack channels, email, online docs, tools
Post-simulation analysis, regression analysis
Synthetic stress scenarios, variations

# Initial reactions

"Is this at all possible?"

"Do you think you can do it?"

"Is it feasible?"

"Wait, we're building one venue, you are going to build 20???"

"Do you understand how much work this is?"

"I would have laughed if anyone else had suggested this"

This convinced us that we were on to something 🙂

# What we aim for

#### Very low resource footprint for simple simulations

Ability to run 100's of simulator instances in one server Ability to start a simulator instance instantaneously

#### Deterministic execution of each simulation

- Capture and storage of **telemetry data** of each simulation
- An extensive library of synthetic test scenarios
- Ability to run as part of a CI-chain
- Automation, automation, automation

# Trading environment





# Simulator architecture

Device Under Test ("DUT")	
	187 0
External access network	
Gateway Gateway Gateway Gateway	
Internal simulator network	
Sequencer Router Matcher Merger Publisher	

## A side-note for the geeks

# Anyone recognize this?



# Intel Skylake microarchitecture



