



# Bring out the big guns

Load testing with  **Gatling**

My slides are / will be available for you at:

<https://bit.ly/3D18xoL>



## Lukas Pradel

### Focus

- Agile Software Development
- Cloud
- Software Testing
- CI/CD & DevOps
- Riding my 



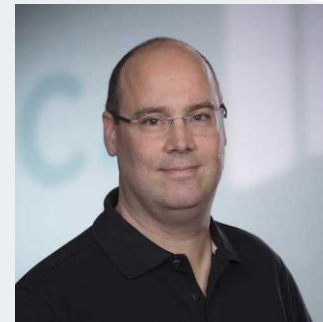
### Contact

- Twitter: @lukaspradel
- eMail: lukas.pradel@conciso.de
- Mobile: +49 160 9034 4938

## Andrej Thiele

### Focus

- Agile Testing
- Agile Software Development
- Domain Driven Design
- Coach for TDD and Agile Development
- Coach for Course “Agile Testing for the Whole Team”



### Contact

- Twitter: @anthie44
- eMail: andrej.thiele@conciso.de
- Mobile: +49 151 1739 7003



# What does Load Testing mean?



Fail Over Test

Performance Test

Peak Load Test

Stability Test

Failure Test

# Load Test

Stress Test

Capacity Test

Overload Test

Recovery Test



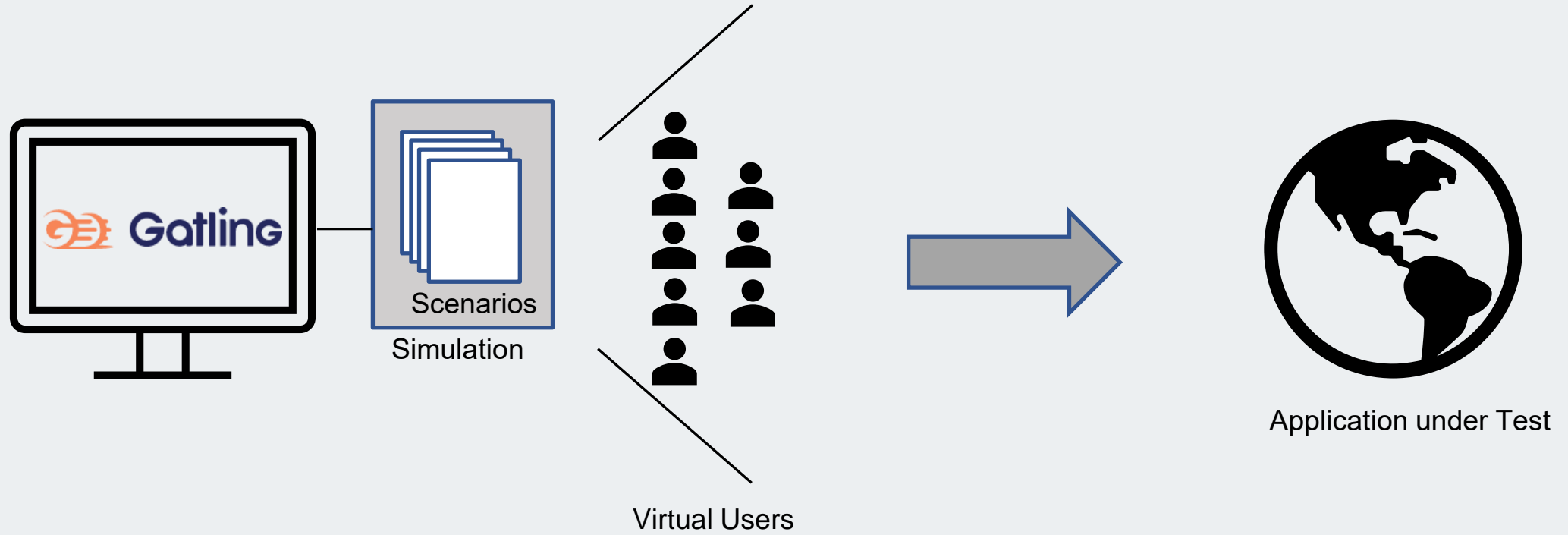
# Goals for Load Testing?



# Preparation?

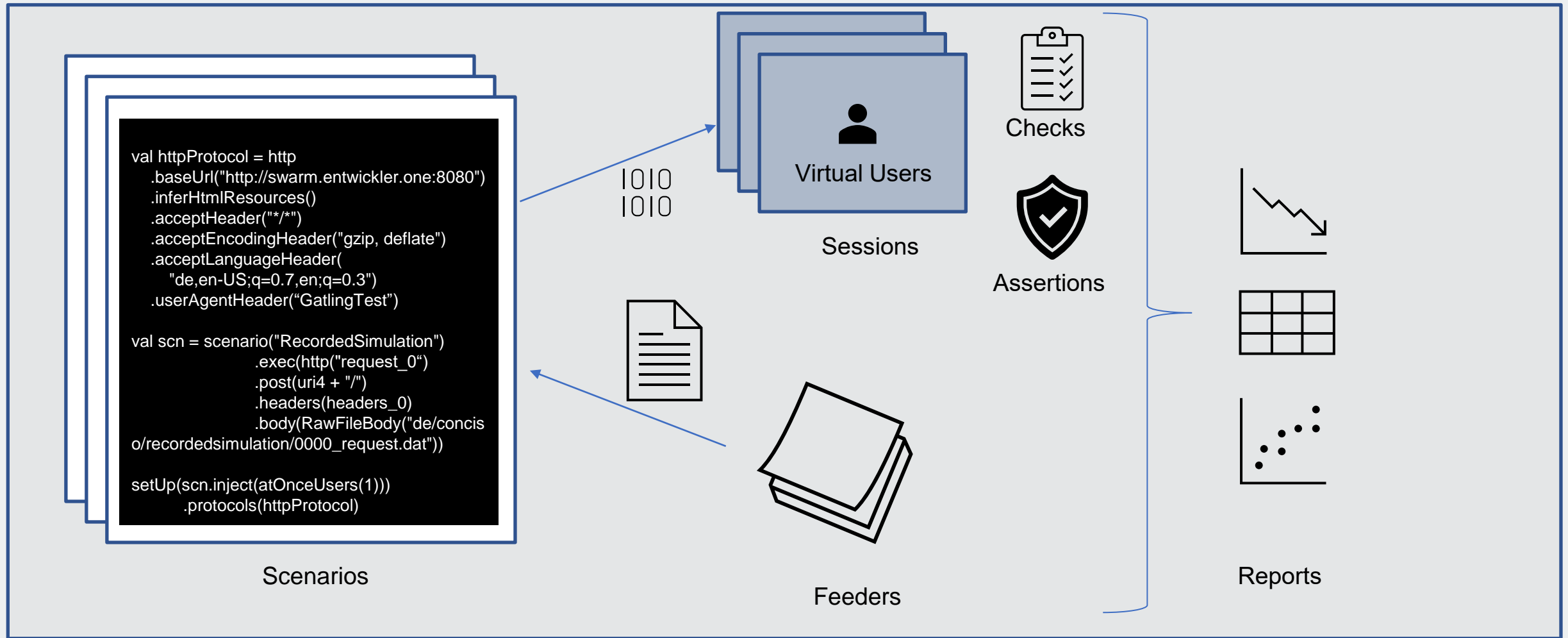


# How does it work?





# How does it work?



Simulation



# Lesson 1 – Prep

*Suggestion: work in pairs or small groups; rotate observers and typist*

Download & install:

- Docker – <https://www.docker.com/products/docker-desktop>
- Scala 2 SDK – <https://www.scala-lang.org/download/>
- Gatling Bundle – <https://gatling.io/open-source/>



# Lesson 2 – Setup

- Download Docker Compose file: <https://bit.ly/3HnpIIR>
- Execute: `docker-compose up`
- In your browser go to: <http://localhost:8080/test>
- Reload a few times and observe
- Execute: `docker-compose down`



# Lesson 3 – Basic Test

- *(optional) download the skeleton test:* <https://bit.ly/3oAQUv2>
- Create a scenario for our app with **1 GET request per second** for a **total of 100 requests**



# Lesson 4 – Gatling reports

- Inspect and review the generated HTML test report



# Lesson 5 – Ramp-Up

- Adjust your scenario with a ramp-up of **5 requests total over 10 seconds** to warm up the service under test



# Lesson 6 – Assertions

- Add an assertion to ensure that **mean response time is less than 1.5 seconds**
- Add an assertion to ensure that the **HTTP response code is always 200**



# Lesson 7 – Fire at will!

- Increase the load in small increments:
- 5 req/s → 10 req/s → 20 req/s → 50 req/s
- How does the service under test behave?
- Can you identify where response time starts to drop?





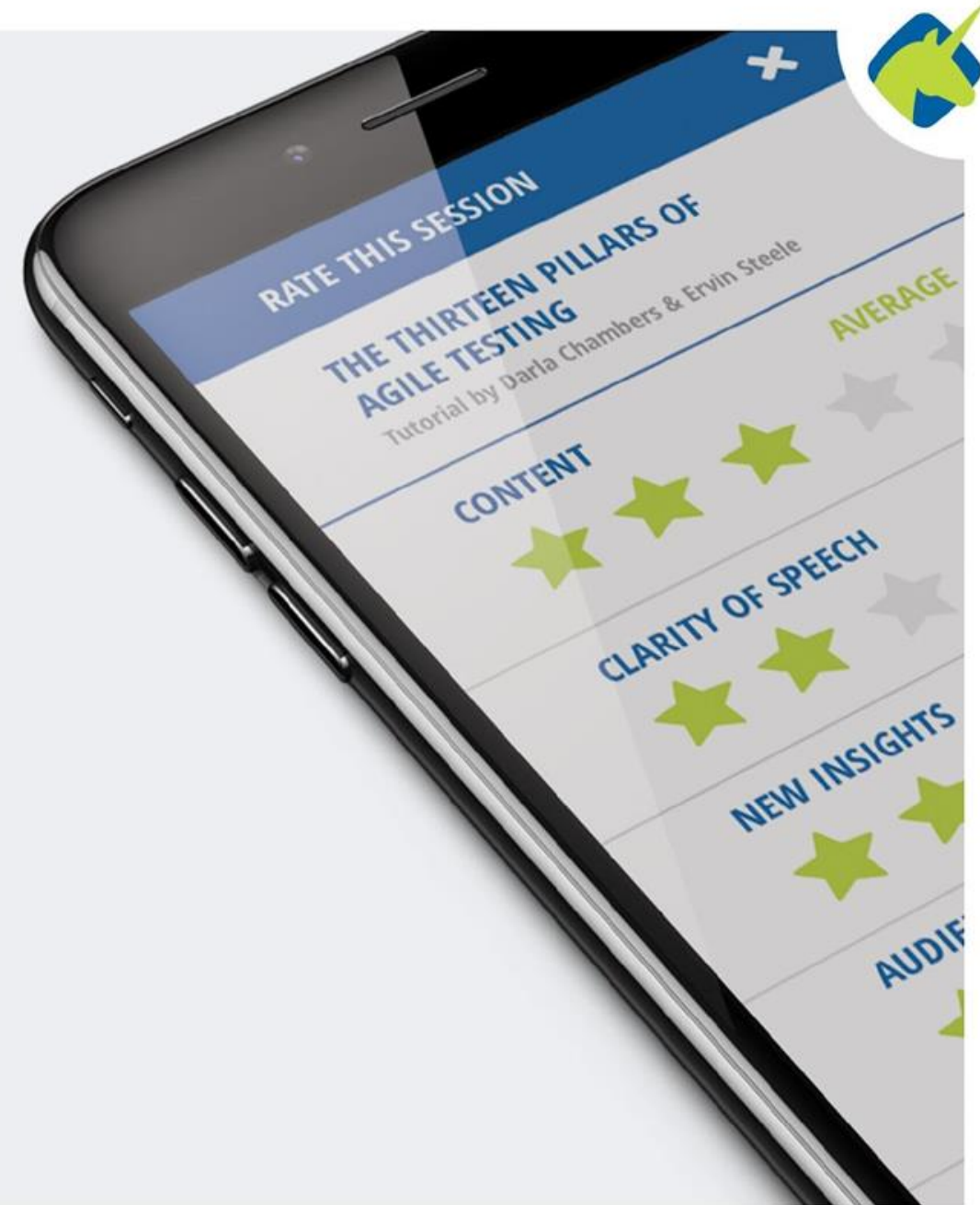
# Lesson 8 – Advanced Topics

- Feeders
- Really heavy load testing (10.000 req/s and more)
- Continuous testing & CI/CD



**Thanks for your attention!**  
Feedback welcome!

Go to [agiletestingdays.com/session-ratings](https://agiletestingdays.com/session-ratings)  
and give your rating!



# CONCISO.



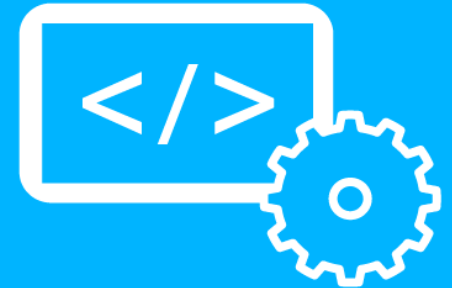
## Agile Advisory

- Agile Training
- Agile Teams
- Agile Organisation



## Viable Architecture

- Architecture Know-how
- Architecture Engineering
- Architecture Evolution



## Pragmatic Development

- Workshops & Trainings
- Coaching & Consulting
- Valueable Software