

# Climbing <sup>↑up and</sup> the pyramid <sub>down↓</sub>

Fredrik Mellgren



# About me

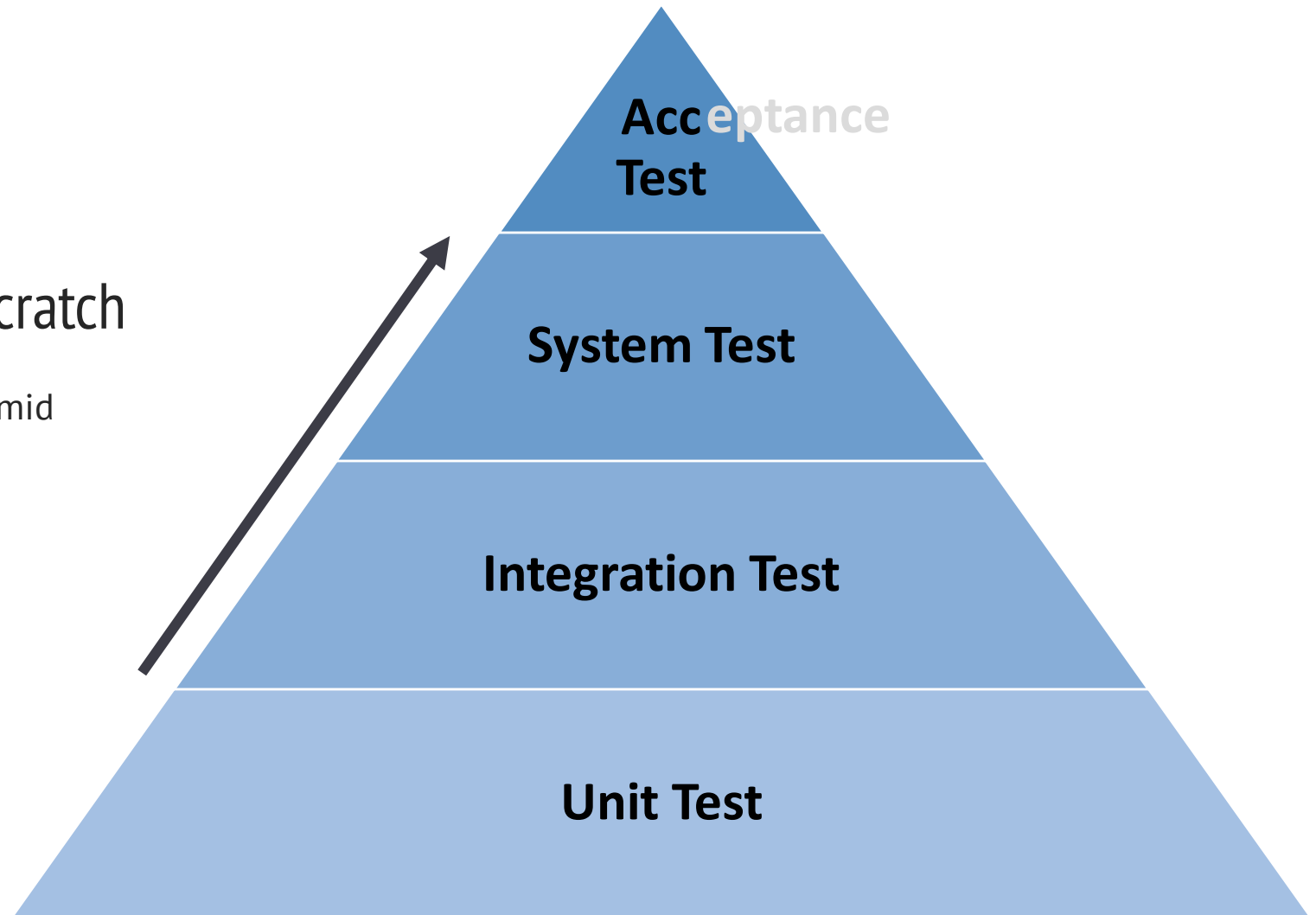
A long time ago, Fredrik was born in Germany where he grew up and studied computer science. Two days after his final presentation, he moved to Warsaw in Poland (yes, because of a girl) and found a job as a project manager. After one and a half years, he married his girlfriend and they moved to Stockholm where Fredrik started working as a Java developer. This was in the end of 2008, a few years later he got interested in test automation and in 2012, he got the opportunity to switch to Rebtel as a test automation engineer. That company

focused a lot on automating all tests and the whole CI/CD flow, so it was an excellent starting point for a test career. Since 2015, Fredrik has been working as a test automation consultant at Regent which has won the award of being Sweden's best working place (according to Great Place to Work) twice in a row. Over the years, he has moved more and more from purely hands-on work towards the QA architect role. He thrives best when he can support others or ventilate thoughts about QA and CI/CD. Which is why he started cooperating with various

vocational schools and gladly speaks on conferences. In 2019, Fredrik started up an own business area called Quality which focuses on test, automation and CI/CD pipelines. And if you've read this far, you are probably working within QA and should consider a career at Regent, drop me a message and I'll be more than happy to discuss this with you. Now we are in the year 2022 and business has grown very quickly. Yes, maybe I've said enough about myself by now and we can start with the actual presentation which you probably are waiting for.

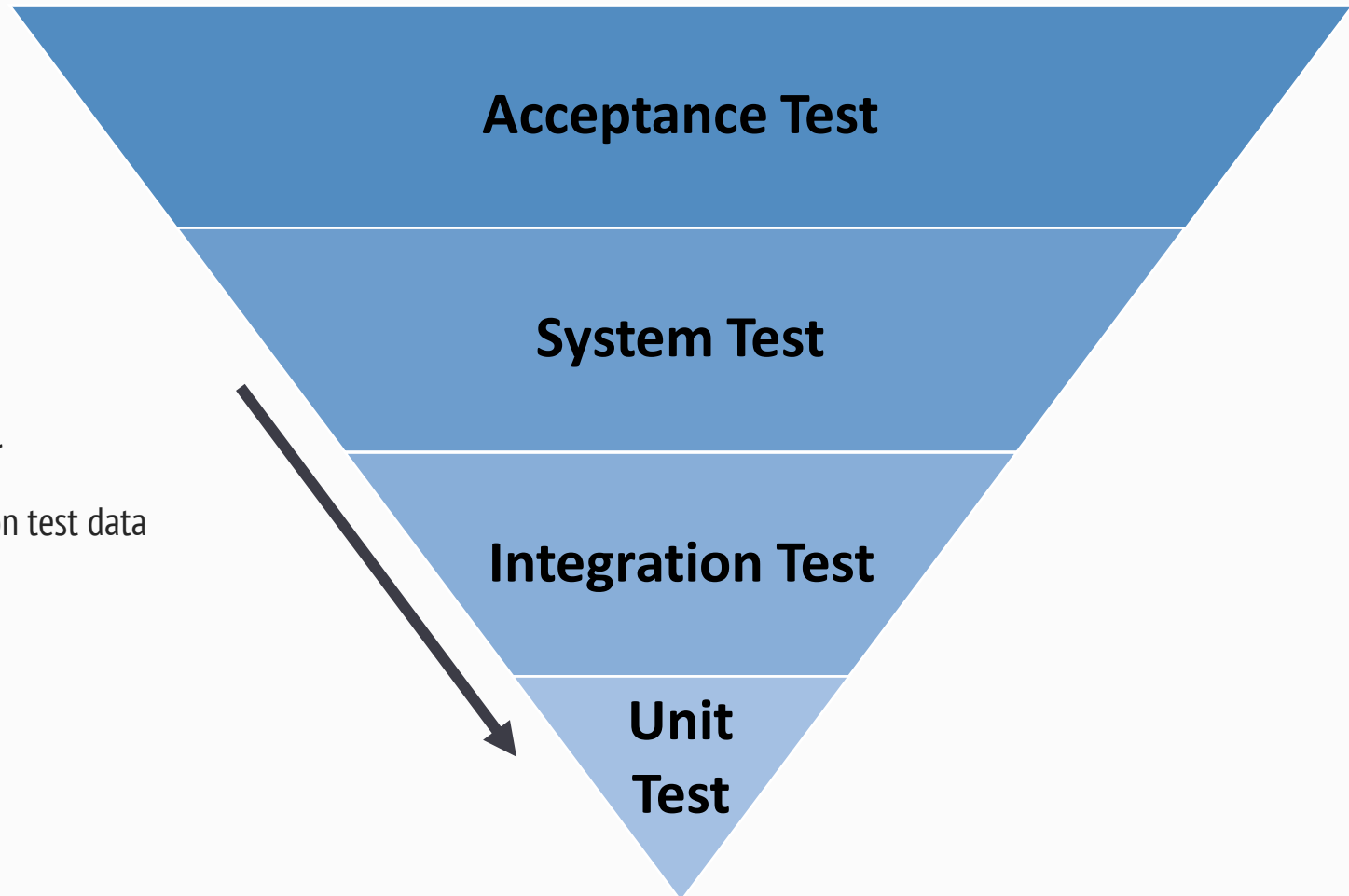
# Climbing Up

- Starting new project from scratch
  - Ideal to work the way up the pyramid
  - TDD

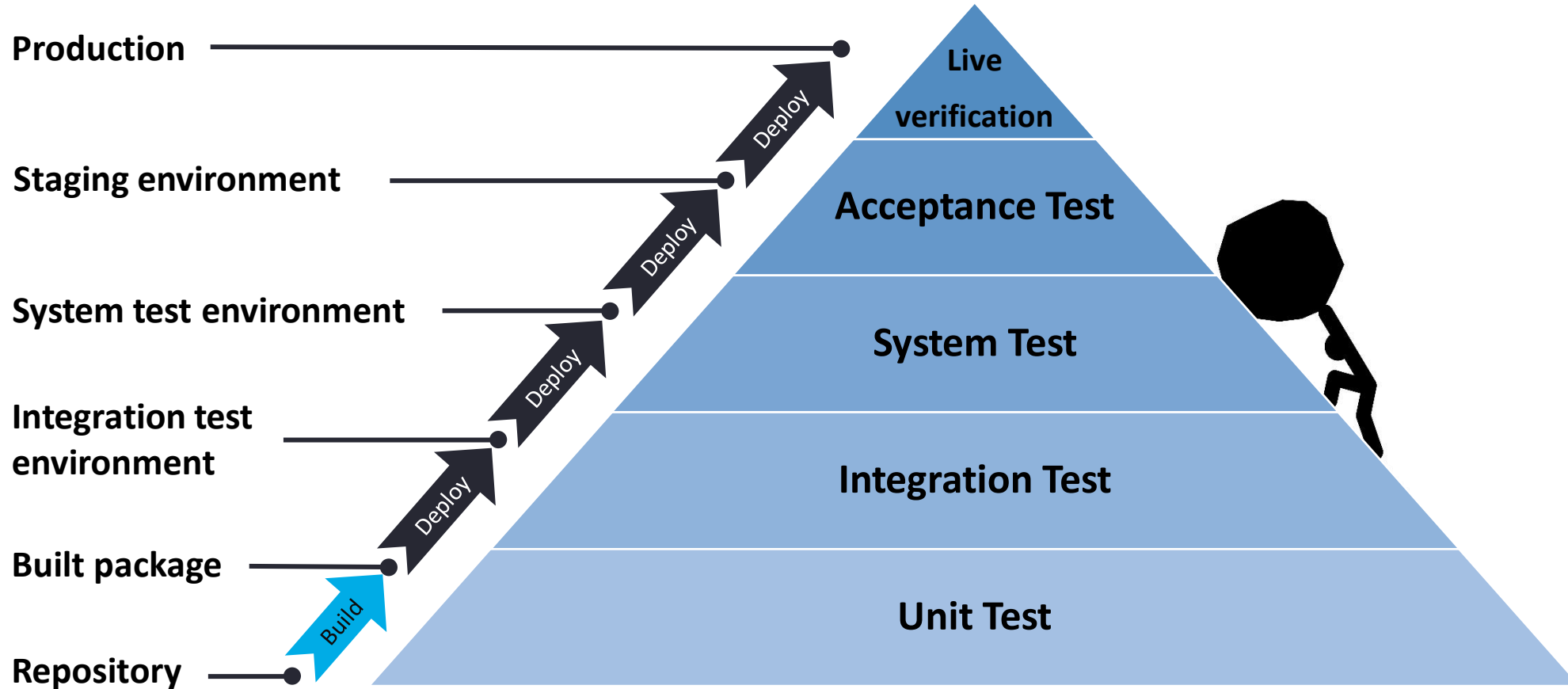


# Climbing Down

- Older projects
  - Often lack automation
  - Mainly testing on system level
    - Instability due to dependency on test data
    - High maintenance costs
    - Shifting down the pyramid
      - or left in the CI/CD-flow

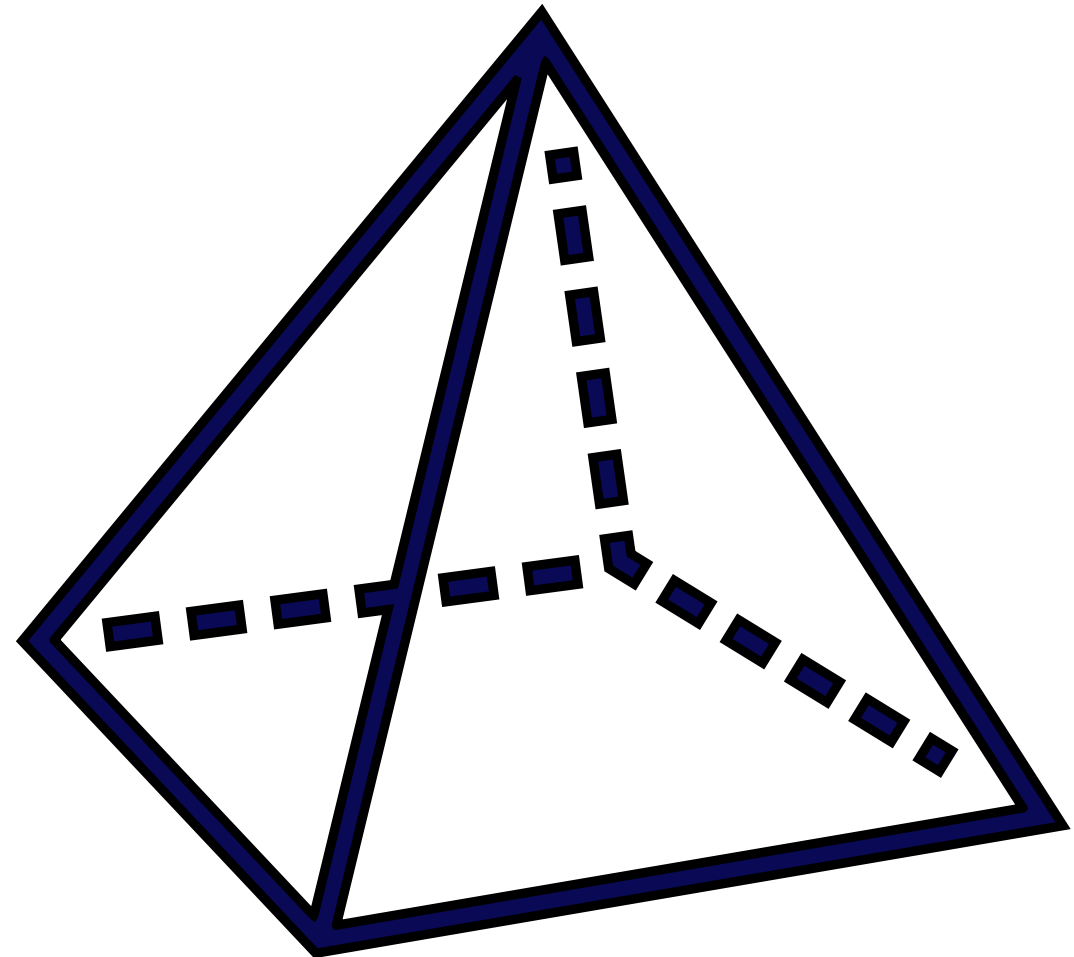


# CI/CD Flow



# Unit Tests

- Easy and fast to test different scenarios
- Lead to better code design
- TDD reduces need for refactoring



# How to get Developers to Unit Test

- Convince management of its importance
- Discuss it at retrospective
- Add unit testing to definition of done on development tasks
- If working with time estimation – developers should add time for unit testing
- Give the developers the right tools – use your network
- Start measuring test coverage and visualize the progress – define goals

# Integration Tests

- Narrow

- Test basic interactions, no flows

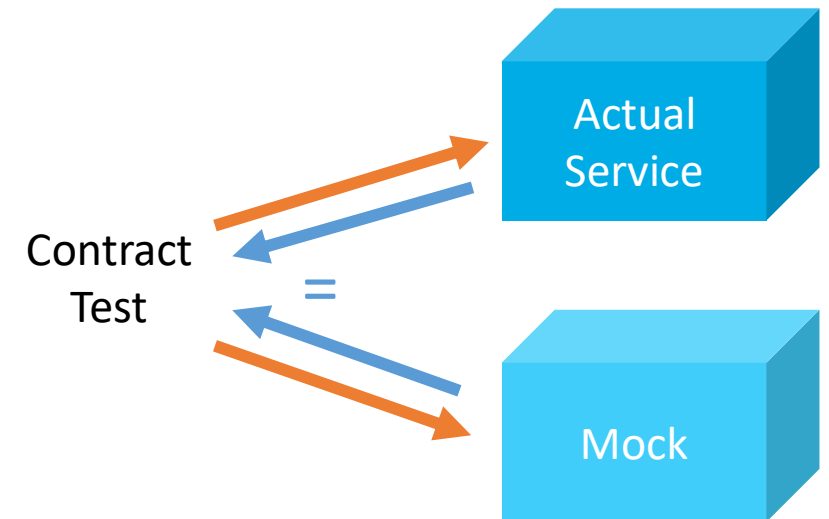


- Wide

- Test short flows within a limited set of components

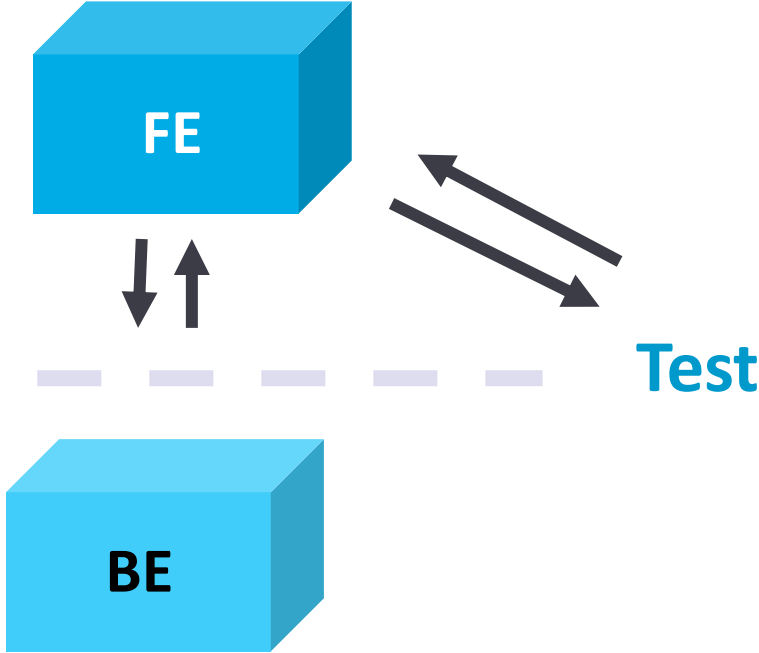
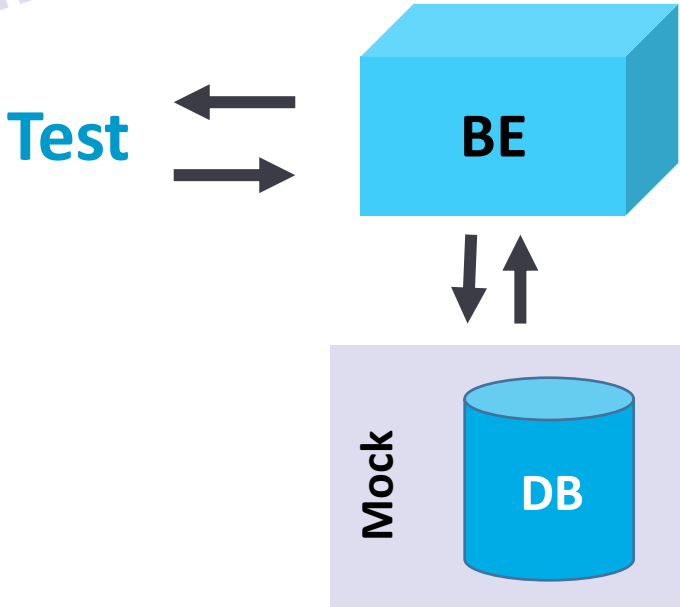
- Contract

- Keep track of changes in APIs we are dependent of
- Check validity of mock data



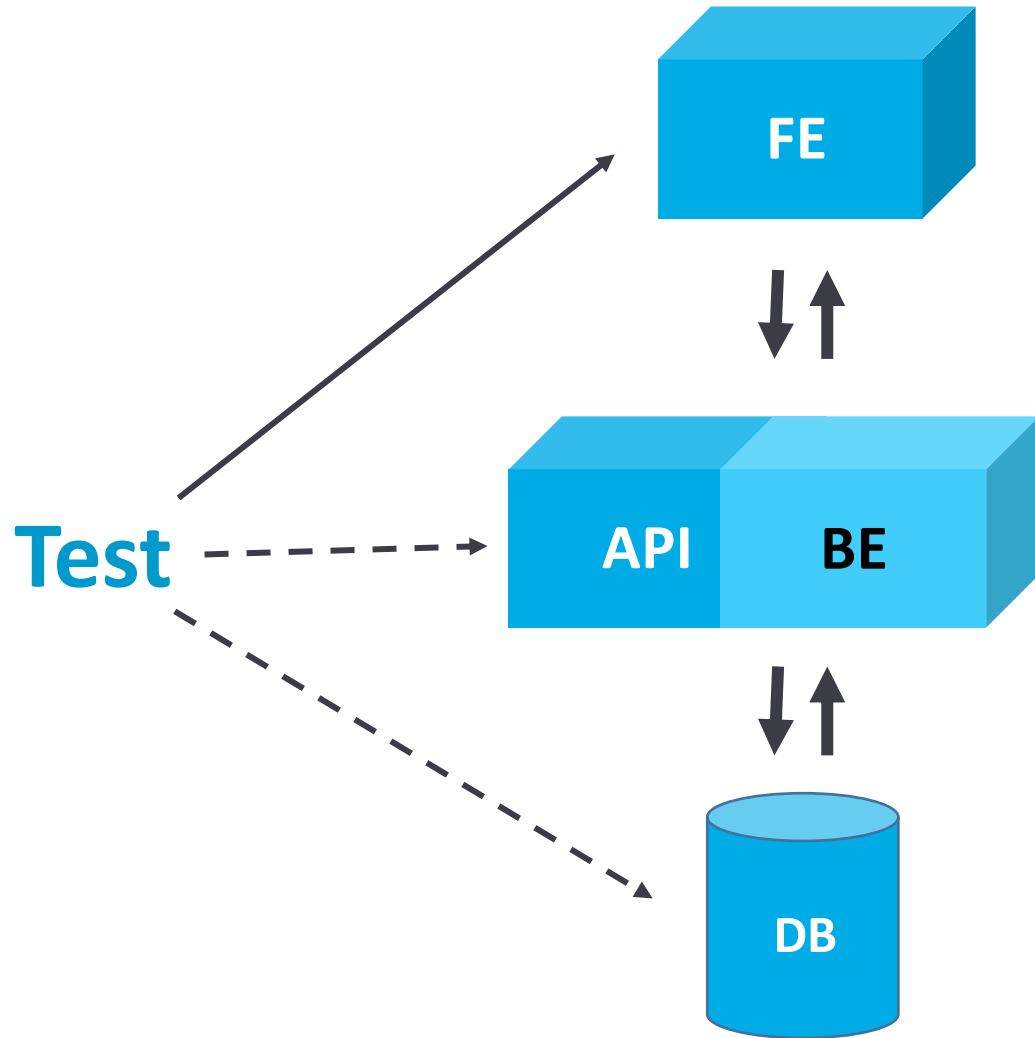


# Integrations to Test



# System Test

- Most important user flows
- Avoid instability
  - Avoid static test data
  - Make use of APIs and databases to find or generate suitable test data
  - Verify metamorphic relations instead of static test data



# Questions?

