

Testing stacked

ALLAN GRAYS APPROACH ON BUILDING QUALITY
SOFTWARE FAST.

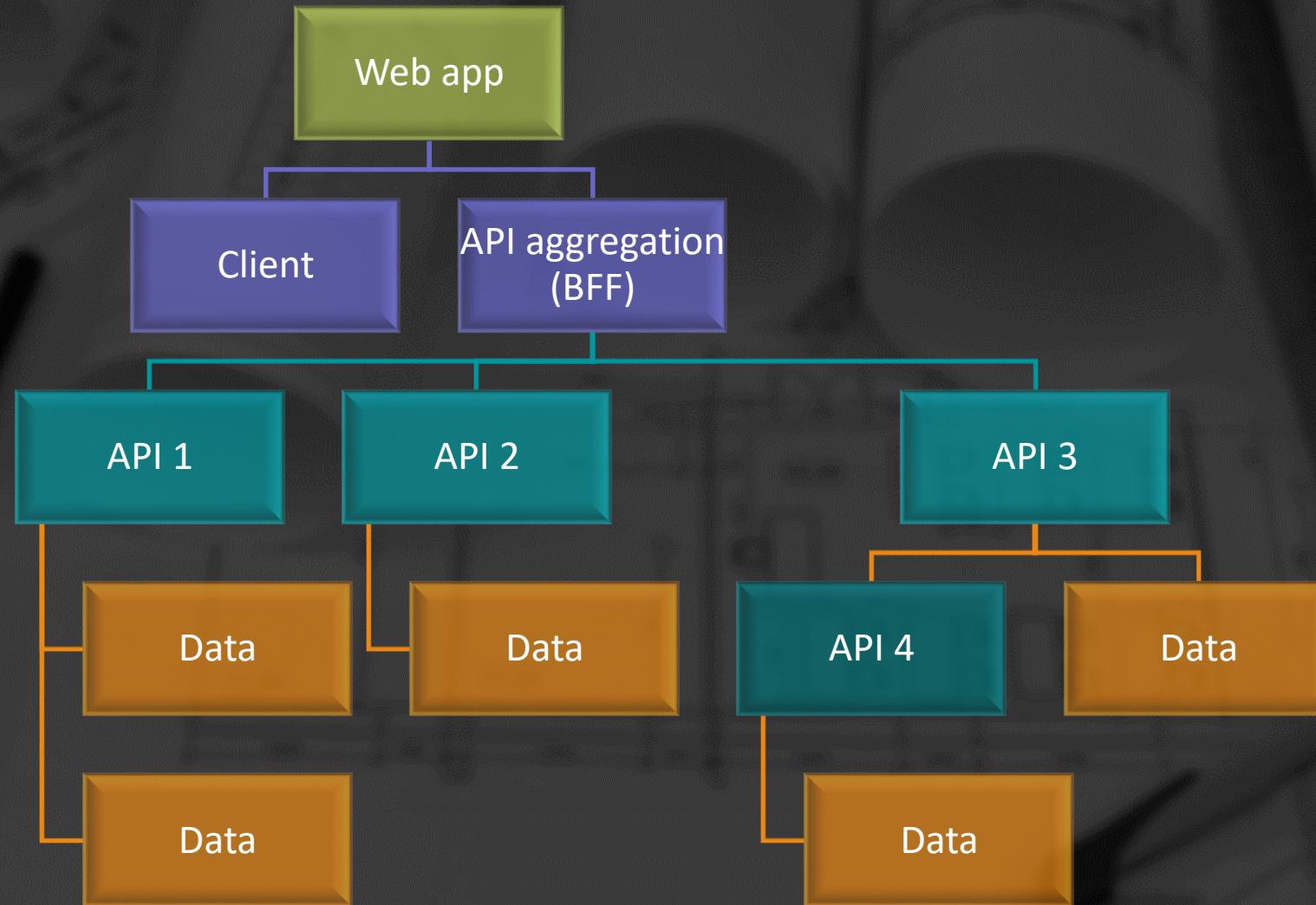
Be zombie proof

- What we really want
- Full control
- Solid
- Unexposed

The balancing act

- Maintaining stability
- Constant change
- Fragile but holding on.

Understanding the architecture



API micro services

- Scala (build)
- Integrated tests with mocked dependencies
- JS (periodically)
- Integrated tests
- Tests that check contracts

API micro services

- Cover basic success and error paths through the integration module.
- Repository connections and error handling.
- The correct data is returned.



IntelliJIDEA

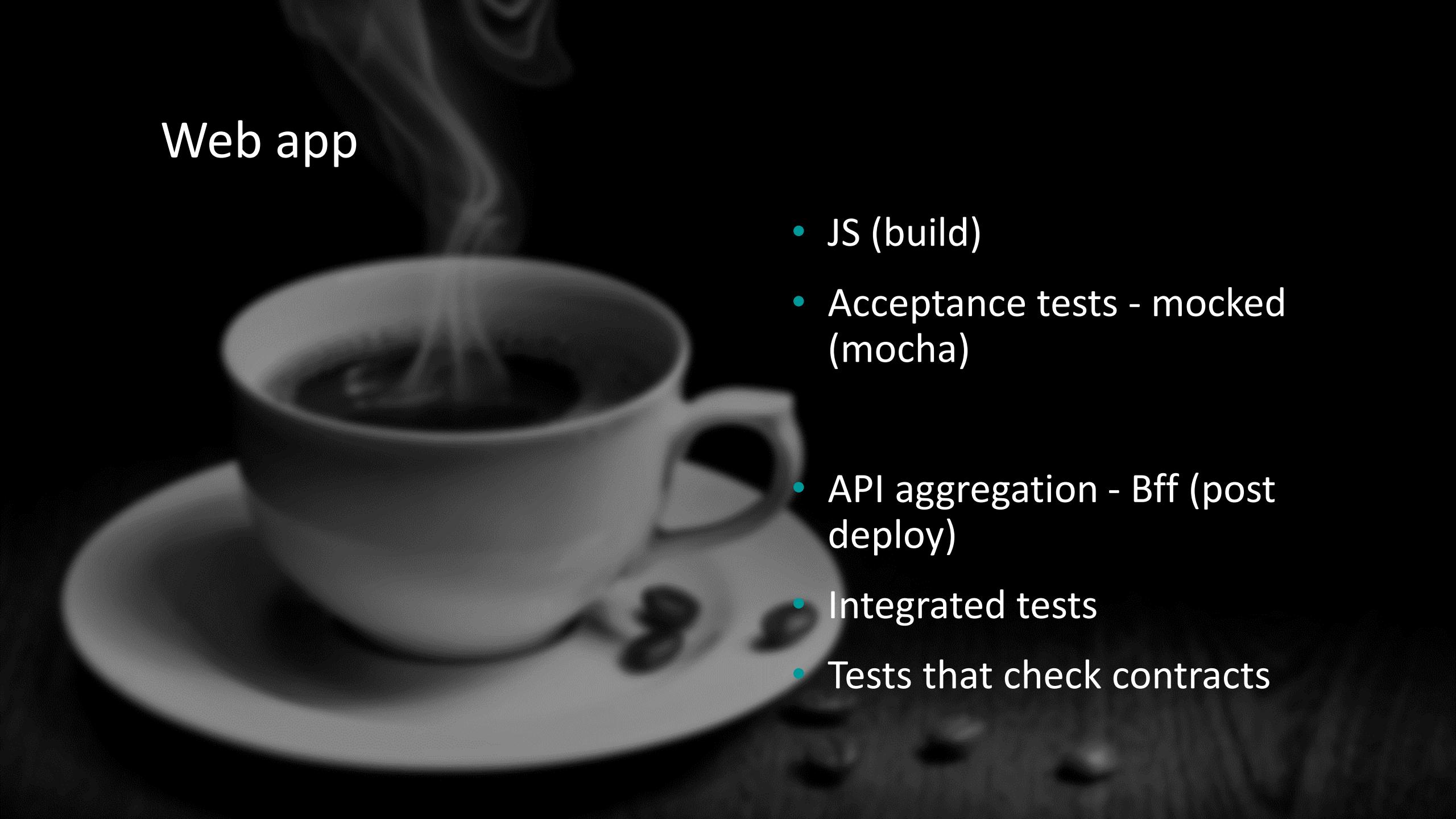


>ScalaTest



GitLab

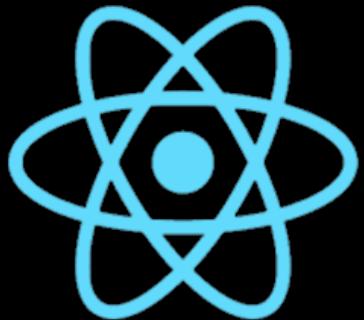
Web app



- JS (build)
- Acceptance tests - mocked (mocha)
- API aggregation - Bff (post deploy)
- Integrated tests
- Tests that check contracts

Web app

- Test all permutations/states of the component.
- User journeys and interactions.
- Fully mocked.
- Runs with build
- Bff integration to independently test service.
- JSON structure and data types are correct.



React

KARMA

mocha



REACCEPTANCE



VirtualBox



NightWatch

- JS (post deploy)
- Via client
- End to End (E2E)
- X-Browser capability
- Multi threaded

NightWatch

- Are all services and apps are talking to each other successfully.
- Quantity: "Happy day" and one or two negative test cases.
- No stubs or mocks.
- E2E tests are there as a second line of test defense.
- Add tests further down.
- E2E tests are expensive.



Nightwatch.js



VirtualBox



Manual verification

REACCEPTANCE



> ScalaTest

Next

- Start a series
- Confusing ideas? Lets talk