Building Spring Boot-based modular libraries for 120 teams

Our findings

Jacek Kaczmarek / Fabio Pezzoni

Spring I/O, Barcelona 18.05.2023



ocado TECHNOLOGY

Offline copy

© Copyright 2023 Ocado Group plc. All rights reserved.

Agenda

- About Ocado Technology
- About our libraries
- How we make it work at scale
  - Working with community
  - Software design
  - Engineering practices

© Copyright 2023 Ocado Group plc. All rights reserved.

Agenda

- **About Ocado Technology**
- About our libraries
- How we make it work at scale
  - Working with the community
  - Software design
  - Engineering practices



© Copyright 2023 Ocado Group plc. All rights reserved.

About Ocado Technology

End-to-end solution for online grocery

- Our solution is called the **Ocado Smart Platform (OSP)**
- It is an **end-to-end suite** that allows grocery retailers to handle all the aspects of the online shopping journey:
  - Managing an online store
  - Fulfilling orders
  - Delivering orders to customers
- **Twelve** global retail partners
  - The solution must be **flexible** and **scalable**
  - **Microservices** architecture
  - Running on **AWS cloud** environments



© Copyright 2023 Ocado Group plc. All rights reserved.

About us



Fabio Pezzoni

Senior Software Engineer

Barcelona (Spain)

Contributor



Jacek Kaczmarek

Senior Staff Software Engineer

Wrocław (Poland)

Maintainer

© Copyright 2023 Ocado Group plc. All rights reserved.

Agenda

- About Ocado Technology
- **About our libraries**
- How we make it work at scale
  - Working with the community
  - Software design
  - Engineering practices



© Copyright 2023 Ocado Group plc. All rights reserved.

*[PragmaticEngProd-Kanat]
**[OcadoTech-Handbook]

The need

Engineering productivity*

- Company standards (formats, interoperability, security)**
- Up-to-date and consistent stack
- Community of practice

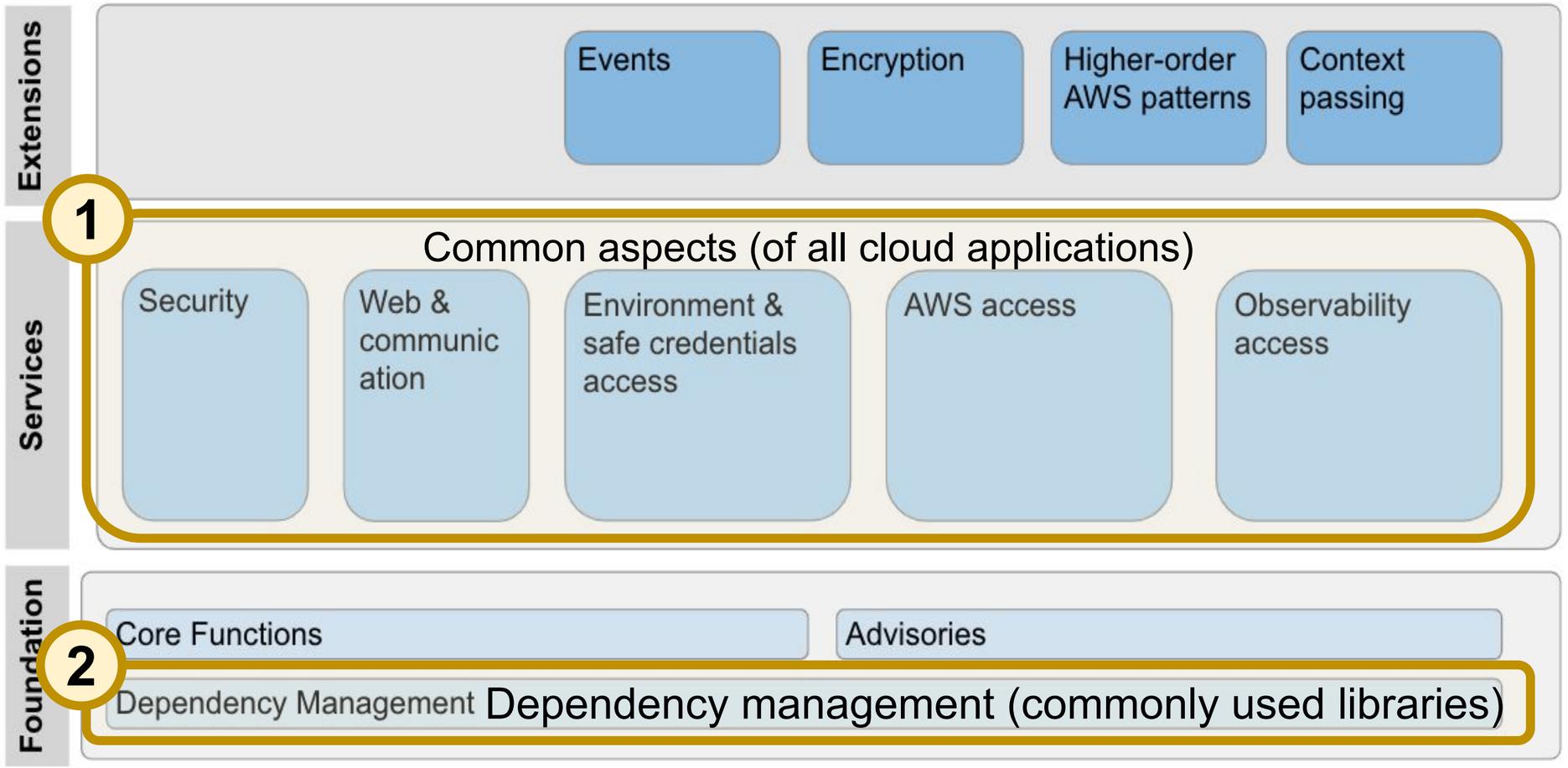
Why these libraries exist?

- Developers created them (inner source)
- Adopted by many teams
- Moved (with team) to Engineering Productivity dept
- Refactored, modularised
- Help users adapt standards and new platform features



© Copyright 2023 Ocado Group plc. All rights reserved.

The scope



more ...

© Copyright 2023 Ocado Group plc. All rights reserved.

In particular

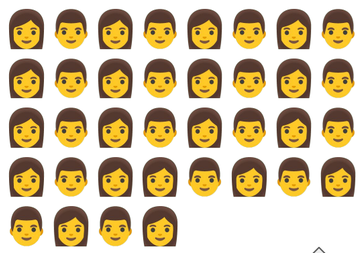
- easy creation of services
- company standards (incl. appsec)*
- up-to-date stack
- interoperability (cloud)

* [OcadoTech-Handbook]

© Copyright 2023 Ocado Group plc. All rights reserved.

The ecosystem

Community

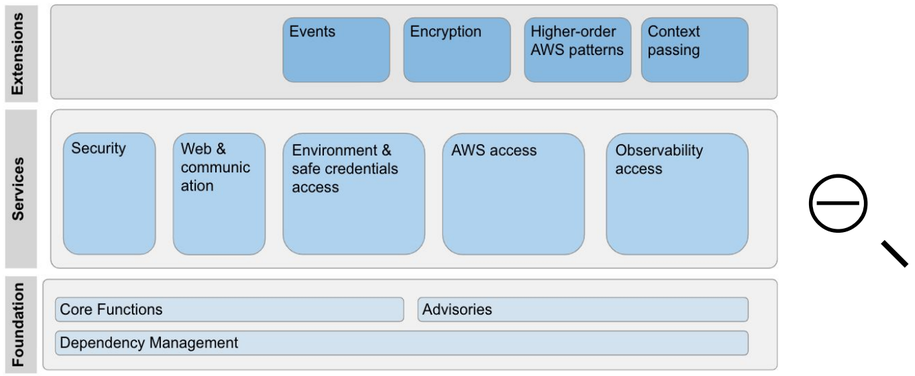


Library #1 • • Library #N

- Ecosystem
- Community of practice (mainly Spring but not only)

Our team





© Copyright 2023 Ocado Group plc. All rights reserved.

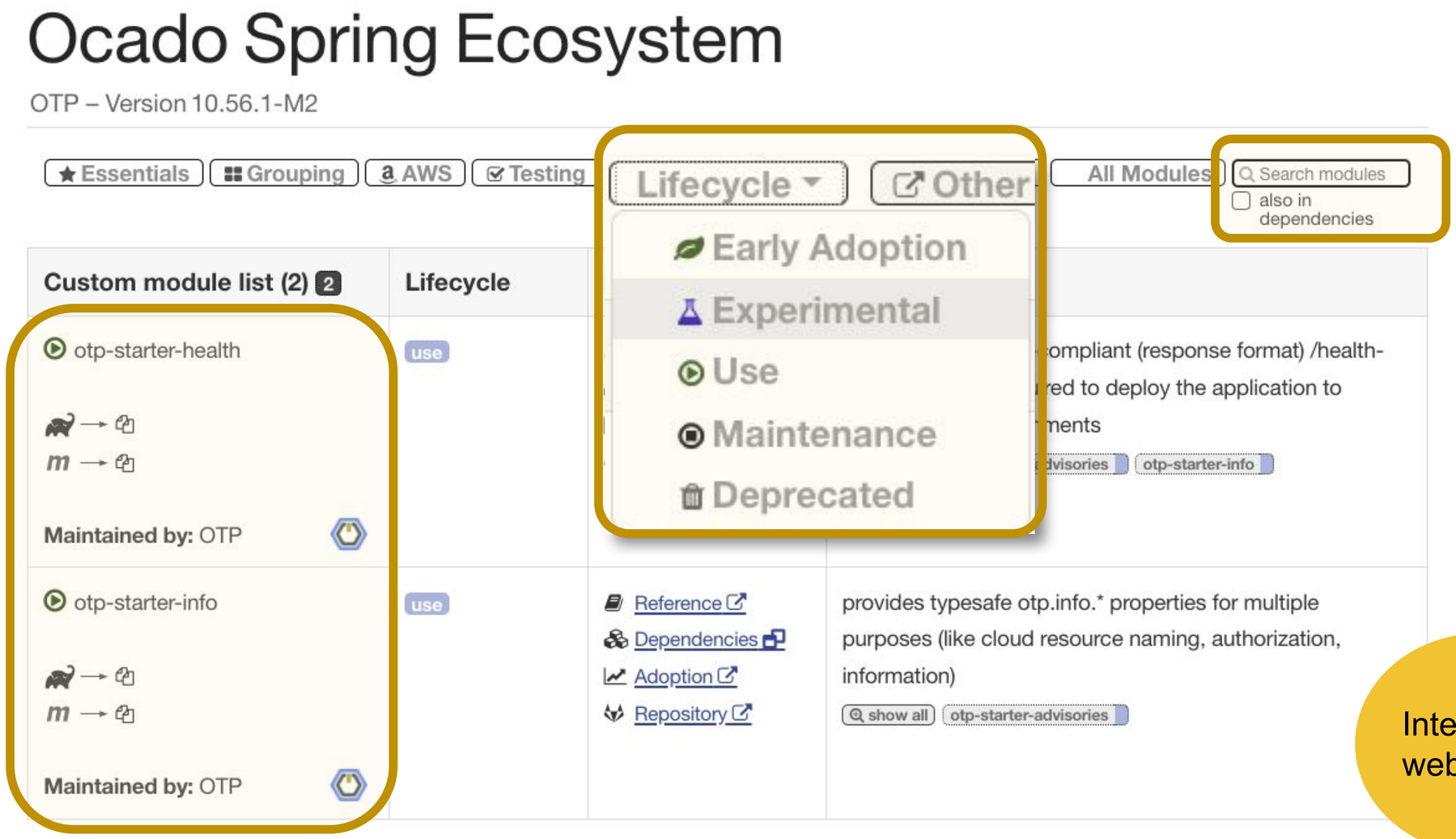
The technologies

- **Technology choices**
  - Spring Boot, Cloud and Cloud AWS

- **Why Spring**
  - Common sense in Java
  - Predictable upgrades
  - Reliable security patching
  - Configurability

© Copyright 2023 Ocado Group plc. All rights reserved.

The form: a shop with modules



Interactive web portal

© Copyright 2023 Ocado Group plc. All rights reserved.

The form: module scope and attributes

#otp-starter-health 1	Lifecycle	Resources	Description
otp-starter-health Maintained by: OTP	use	Reference Dependencies Adoption Repository	provides a hallmark-compliant (response format) /health-check endpoint required to deploy the application to Ocado cloud environments show all otp-starter-advisories otp-starter-info
otp-starter-context-passing Maintained by: OTP	experimental	Reference Dependencies Adoption Repository	module responsible for passing of context attributes between applications through typical integrations (logs, messaging, web) show all otp-starter-advisories otp-starter-info

https://jarjar.pages.tech.lastmile.com/modular-reference/#jarjar-starter-

© Copyright 2023 Ocado Group plc. All rights reserved.

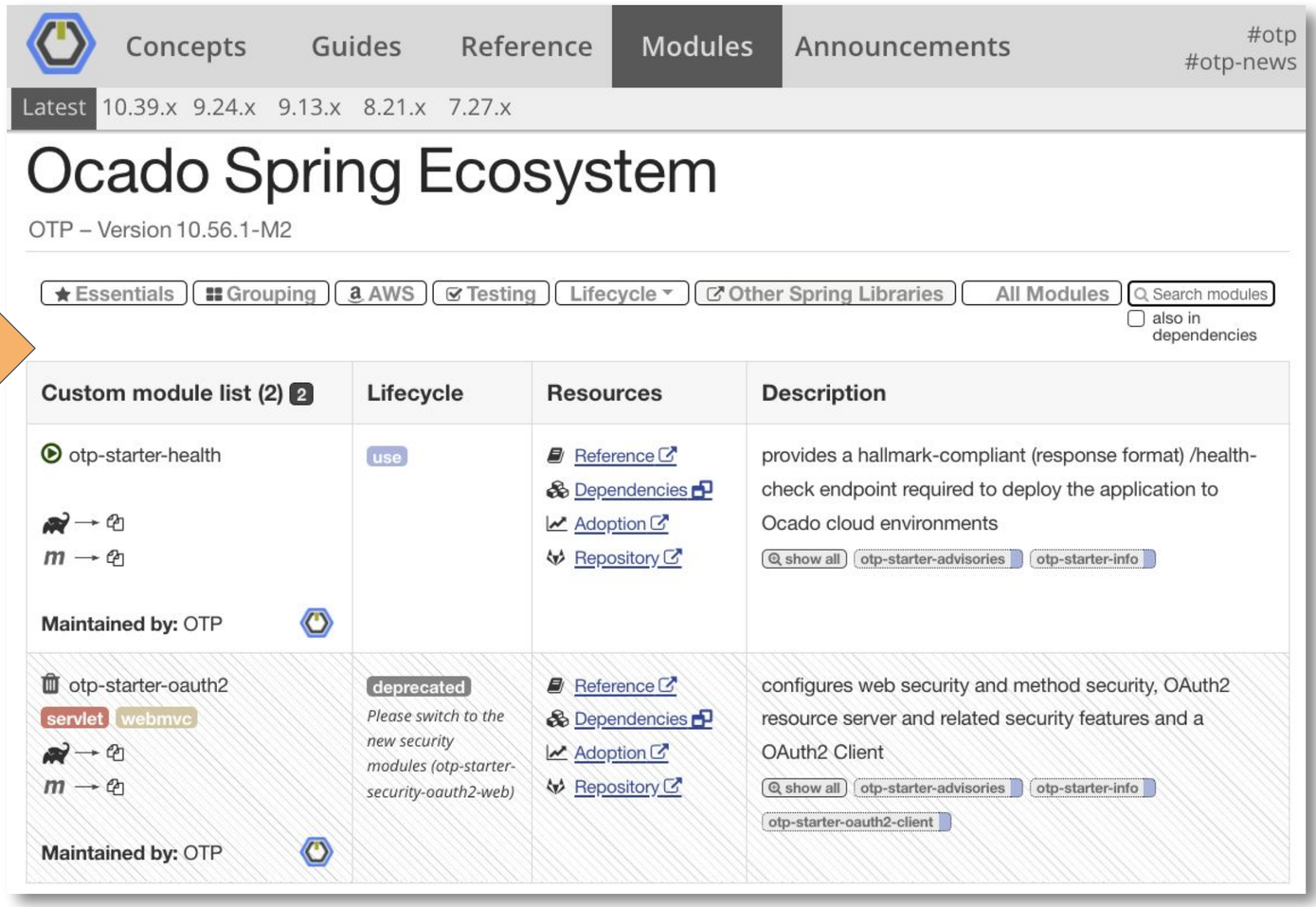
The form: living documentation!

It looks dynamic and complex ..

but is mostly generated

`pom.xml`

`project-metadata.properties`



© Copyright 2023 Ocado Group plc. All rights reserved.

The user build

```
plugins {
    id 'org.springframework.boot' version "3.0.4"
    (...)
}

//Reuse Spring Boot, Cloud, Cloud AWS, AWS SDK, a few internal libraries, patches
implementation platform("com.ocadotechnology.otp:otp:${otpVersion}")

implementation 'org.springframework.boot:spring-boot-starter-web'
...

//configures web OAuth2 and security headers
implementation 'com.ocadotechnology.otp:otp-starter-security-oauth2'
//add /health-check in agreed format
implementation 'com.ocadotechnology.otp:otp-starter-health'
```

© Copyright 2023 Ocado Group plc. All rights reserved.

The user code

- User writes code in Spring and is in control
- Our library is often invisible
- Adding / removing our modules is easy

```
@RestController //plain Spring Boot
public class WebApp {

    private final Service service;

    @RolesAllowed("role")
    @GetMapping("/endpoint")
    public BusinessObject getEndpoint() {
        return service.logic();
    }
...
```

```
@FeignClient(value = "backend")
public interface BackendClient {

    @CircuitBreaker(name = BACKEND)
    @GetMapping("/backendLogic")
    BackendDTO getBackendLogic();

    ...
}
```

Figures

Scope

- **90** modules (starters)
- **200** releases per year*

Scale

- **120+** user teams
- Growing adoption (despite being opt-in)

*we are on **Boot 3.0.6**, Cloud 2022.0.2 (latest version available as of this presentation date 18.05.2023)

© Copyright 2023 Ocado Group plc. All rights reserved.

Agenda

- About Ocado Technology
- About our libraries
- **How we make it work at scale**
  - Working with the community
  - Software design
  - Engineering practices



© Copyright 2023 Ocado Group plc. All rights reserved.

How do we make it work for 120 teams

“It depends” ;)

We observe community and:

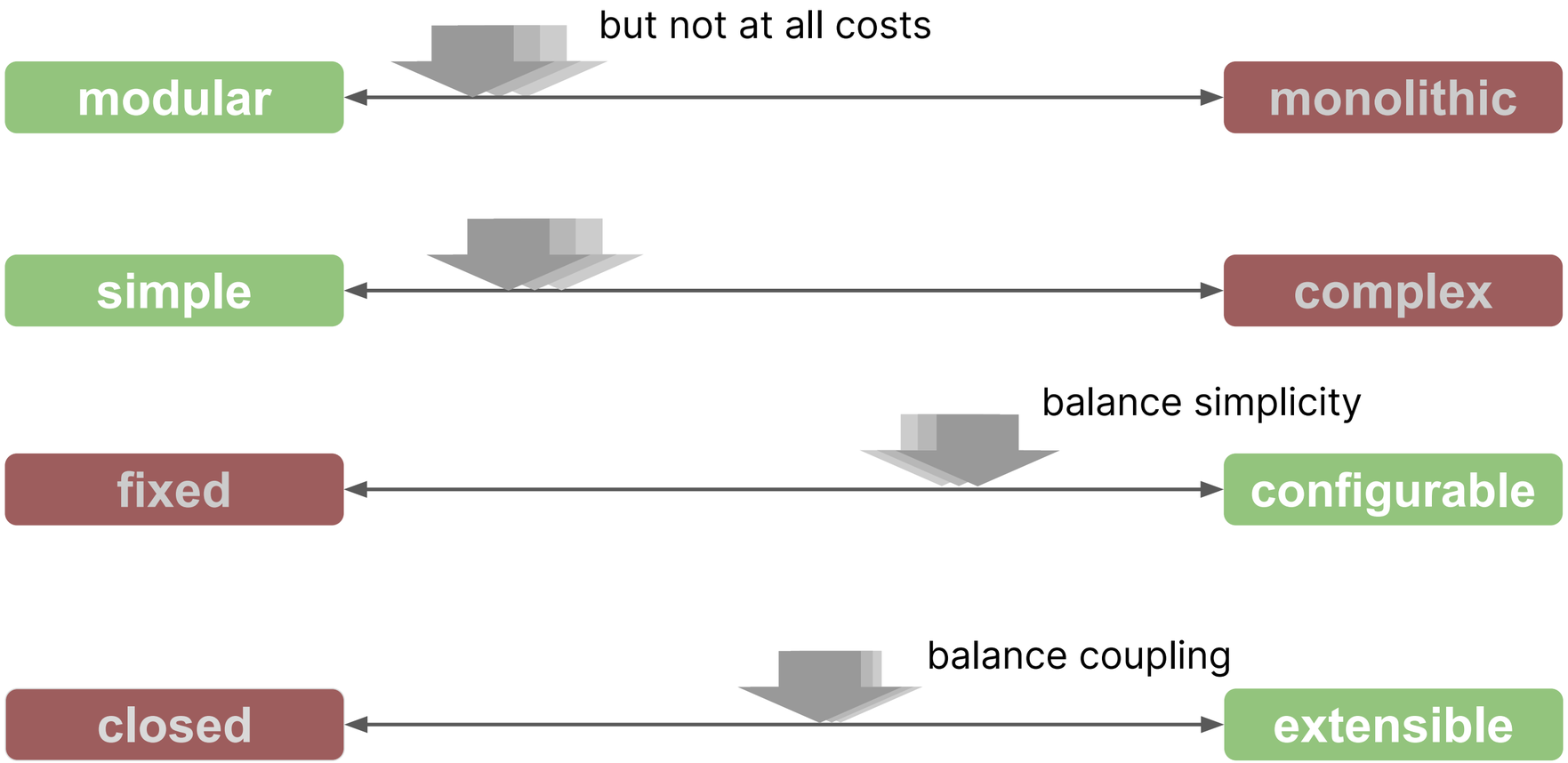
- balance of characteristics over time
- *...ilieties**



*Characteristics that follow are inspired by [Richards-Ford]

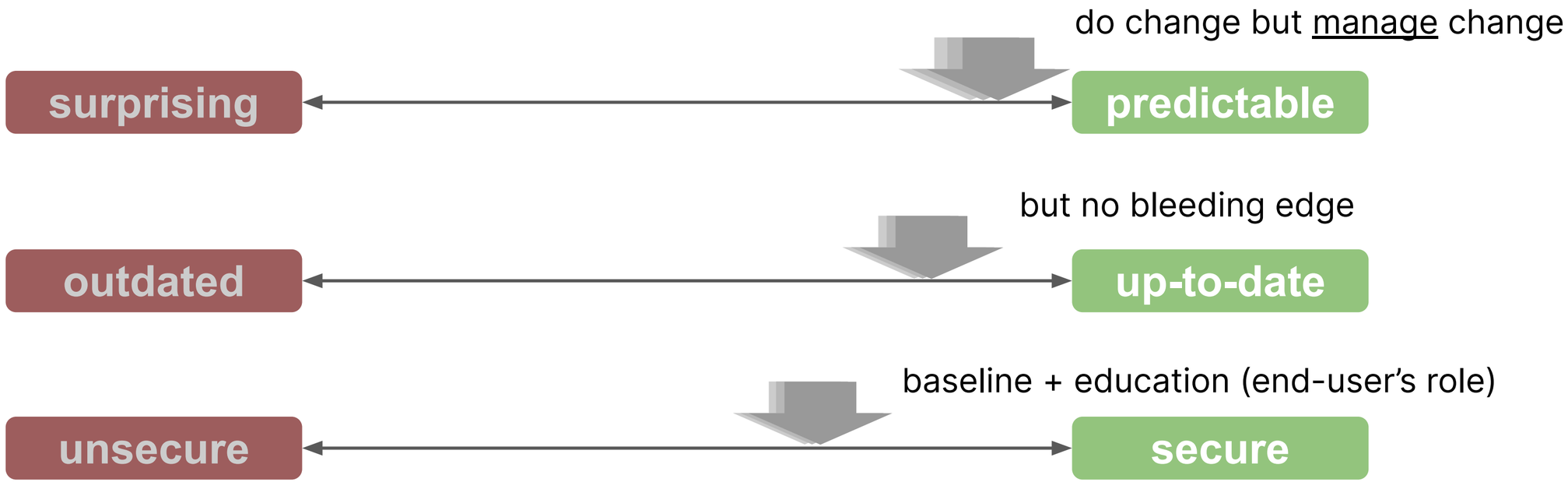
© Copyright 2023 Ocado Group plc. All rights reserved.

Characteristics mixer - design



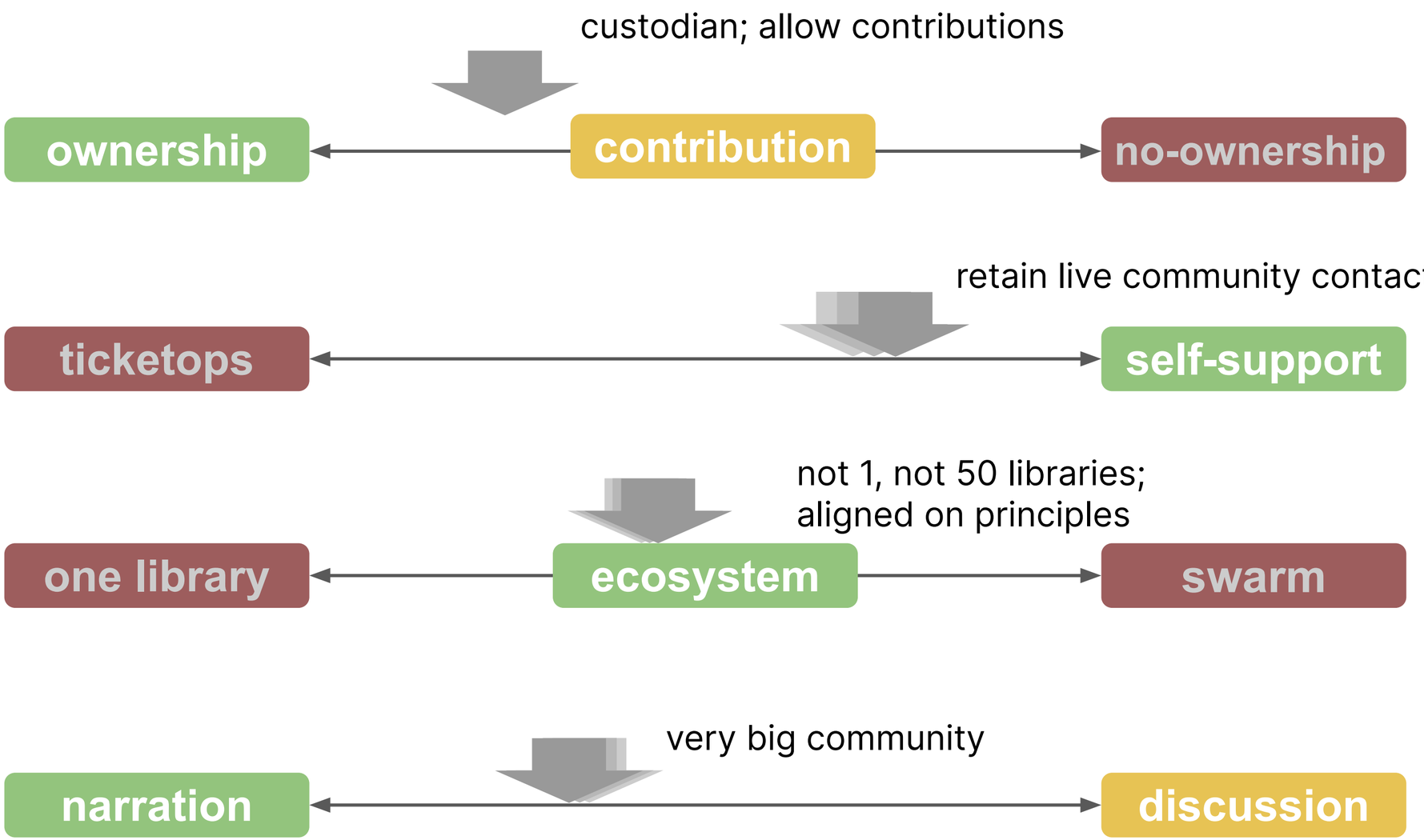
© Copyright 2023 Ocado Group plc. All rights reserved.

Characteristics mixer - dynamics



© Copyright 2023 Ocado Group plc. All rights reserved.

Characteristics mixer - community



[Fowler-SharedLib]

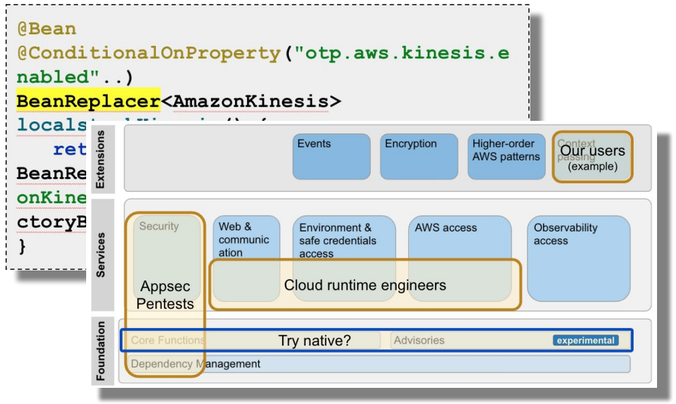
© Copyright 2023 Ocado Group plc. All rights reserved.

How we make it work at scale

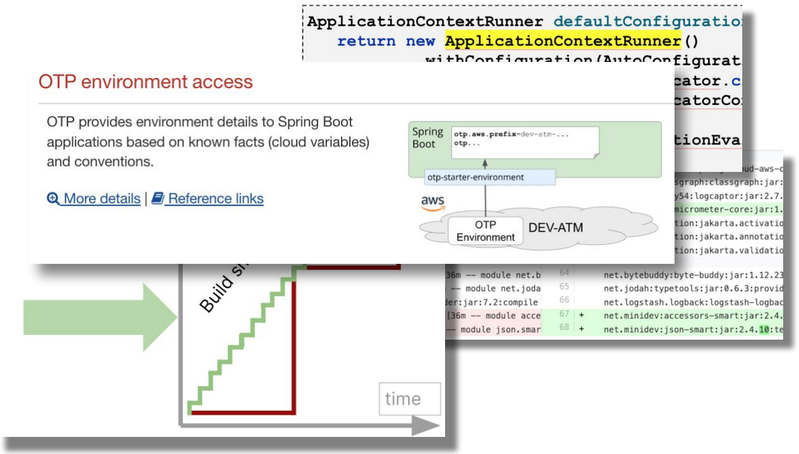
Working with the community



Software design



Engineering practices



Holistic approach

© Copyright 2023 Ocado Group plc. All rights reserved.

Agenda

- About Ocado Technology
- About our libraries
- How we make it work at scale
  - **Working with the community**
  - Software design
  - Engineering practices



© Copyright 2023 Ocado Group plc. All rights reserved.

Working with the community

Big scale of adoption

- 350+ applications in production
  - in 20+ production environments
- 85% adoption at Ocado Technology
- 30% growth last year

Big community of users

- 800+ developers
  - In 120+ teams
  - In 5 departments

Big challenge

- How to manage such a big community of users

Our approach

1. Being custodians
2. Analysing usage data
3. Interacting with people
4. Creating narration

© Copyright 2023 Ocado Group plc. All rights reserved.

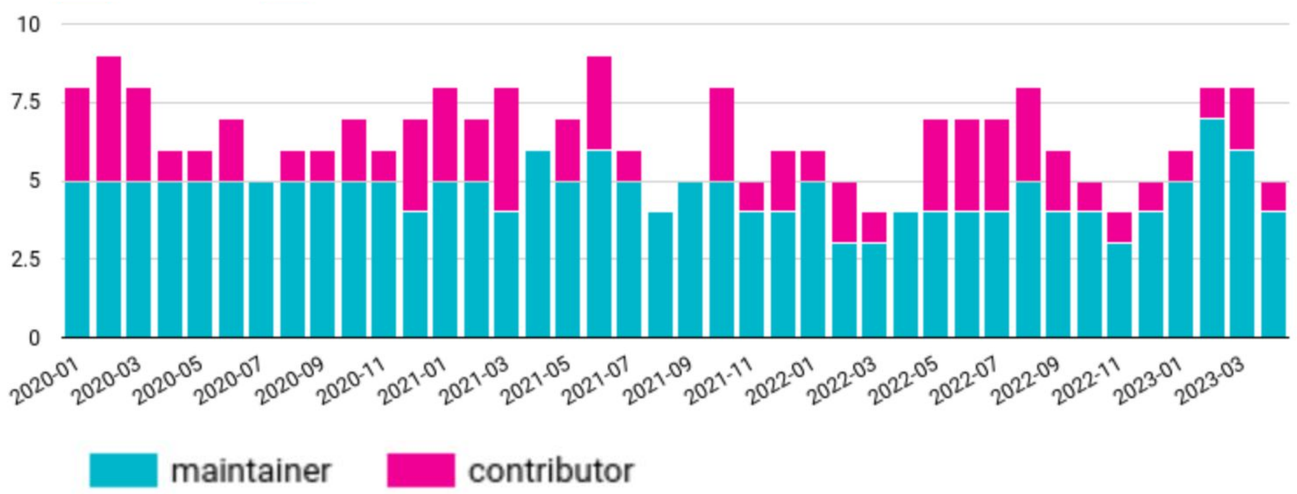
1. Being custodians

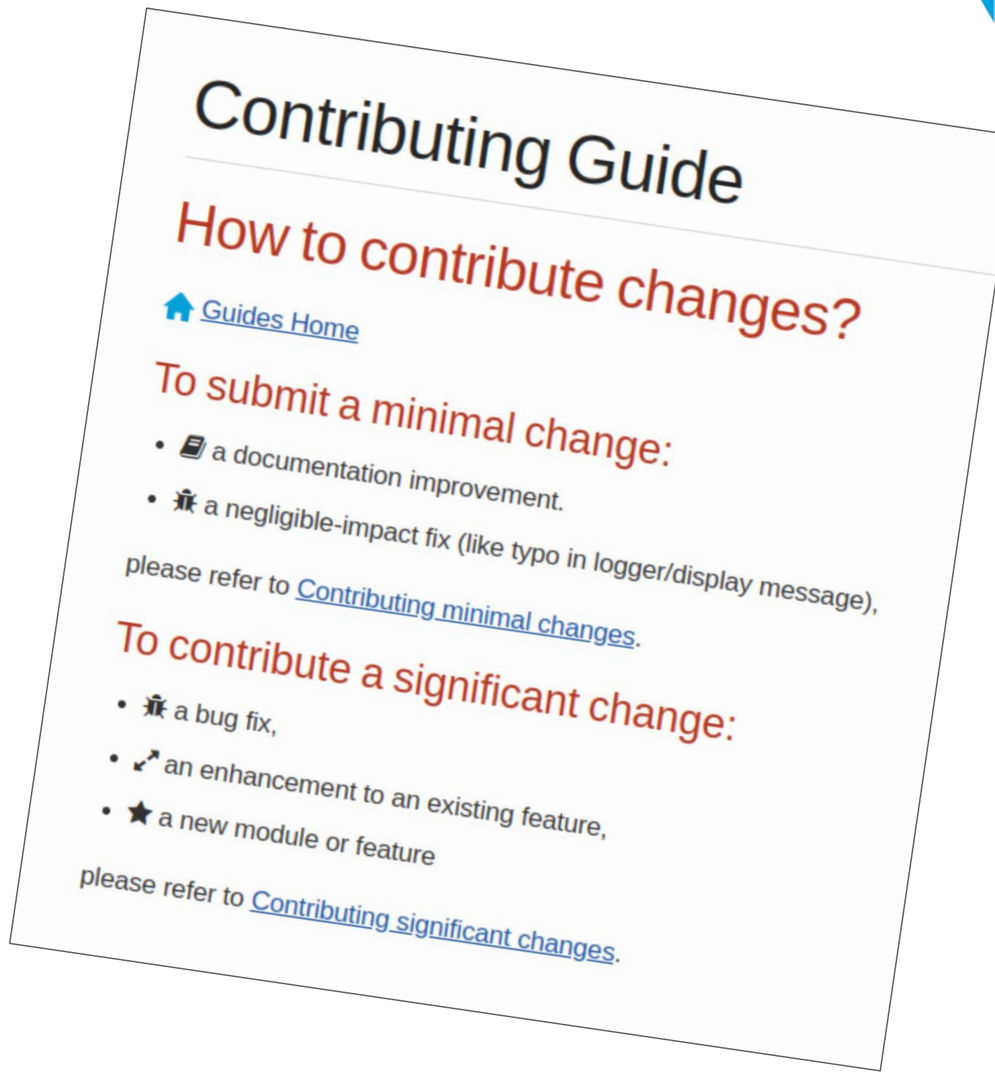
Have the ownership allowing contributions

- Users are encouraged to contribute changes to the libraries
- Different processes for minimal and significant changes

5 maintainers (average)

132 contributors (absolute)





Contributing Guide

How to contribute changes?

Guides Home

To submit a minimal change:

- a documentation improvement.
- a negligible-impact fix (like typo in logger/display message),

please refer to Contributing minimal changes.

To contribute a significant change:

- a bug fix,
- an enhancement to an existing feature,
- a new module or feature

please refer to Contributing significant changes.

© Copyright 2023 Ocado Group plc. All rights reserved.

2. Analysing usage data

Data is fundamental

- For supporting the decision process
- For users discovery

Automatic data collection

- Applications produce data runtime
  - Libraries version, activated features, java version, team info, etc.
- Data collected in live dashboards
  - Aggregated data, trend over time and other statistics
  - Filters by period of time, department, libraries version, activated features, etc.

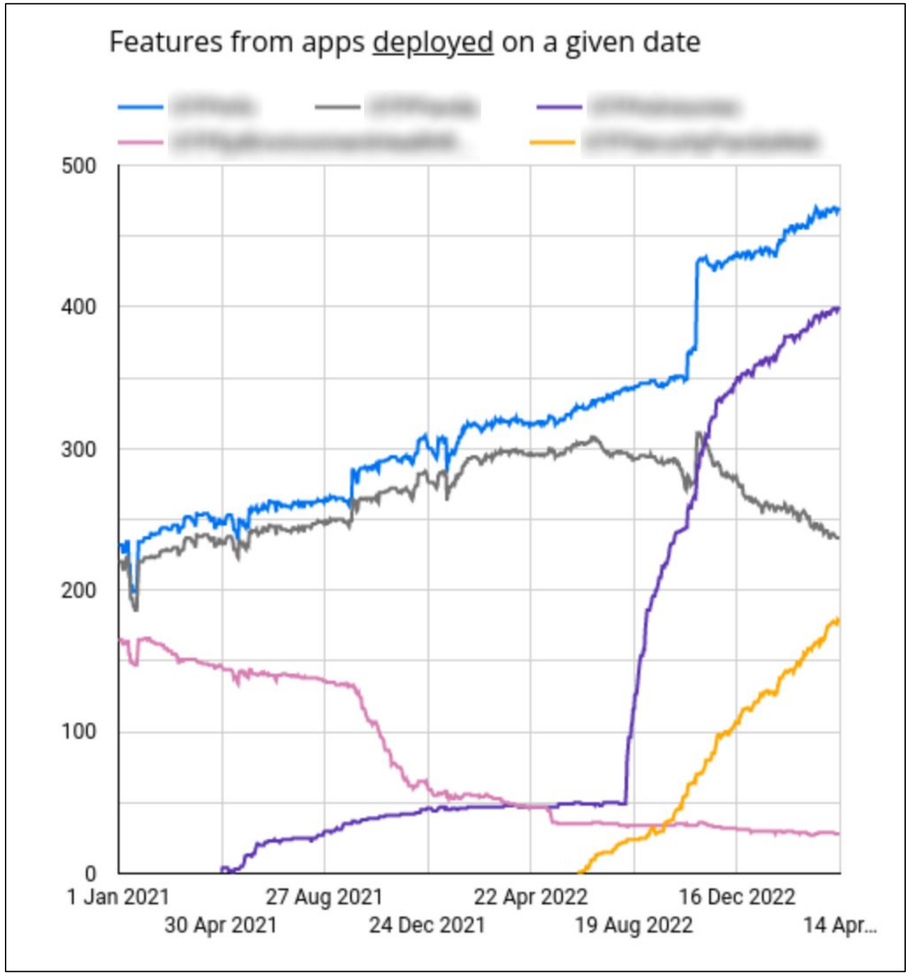


© Copyright 2023 Ocado Group plc. All rights reserved.

2. Example: supporting the decision process

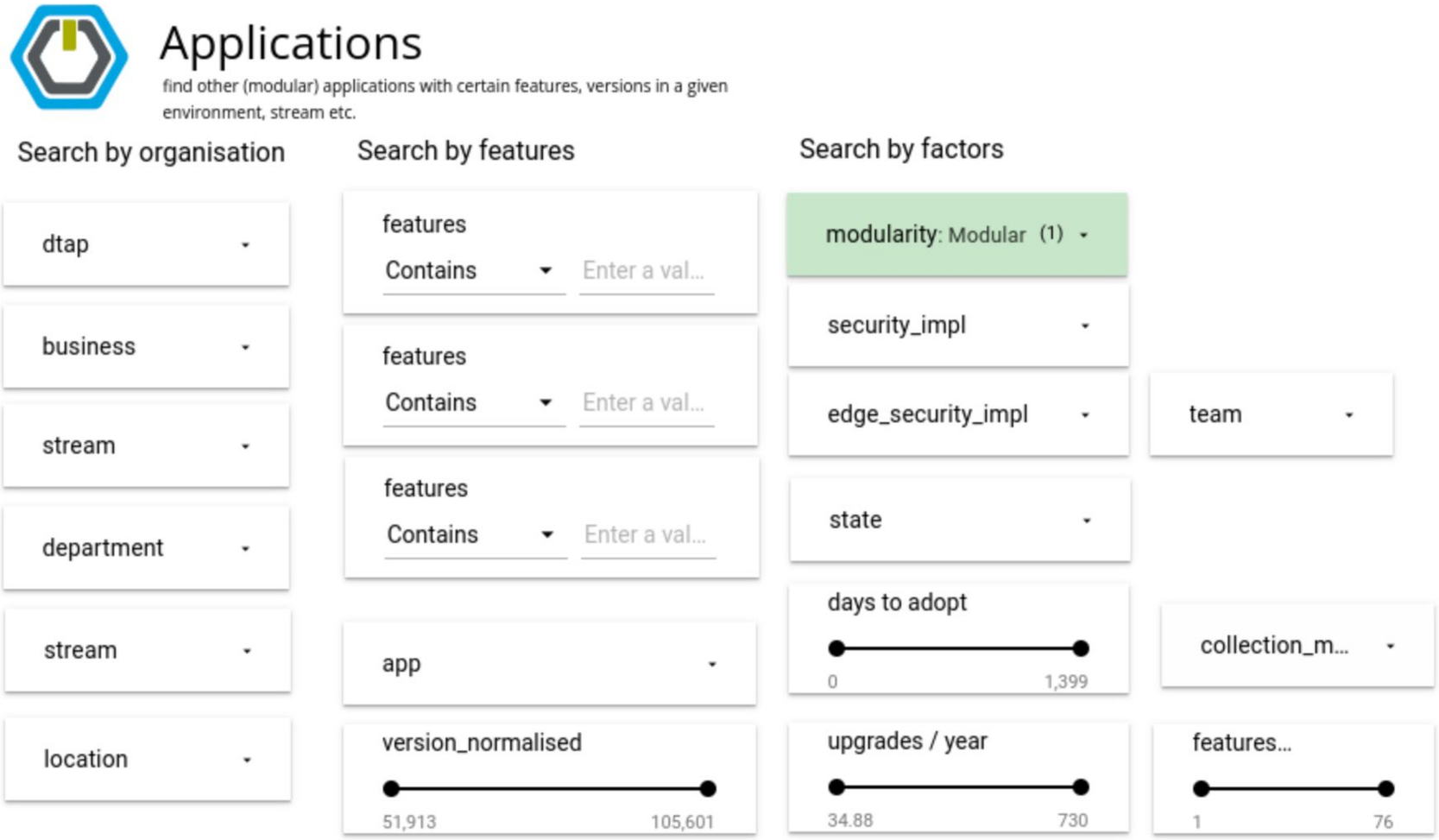
Feature adoption over time

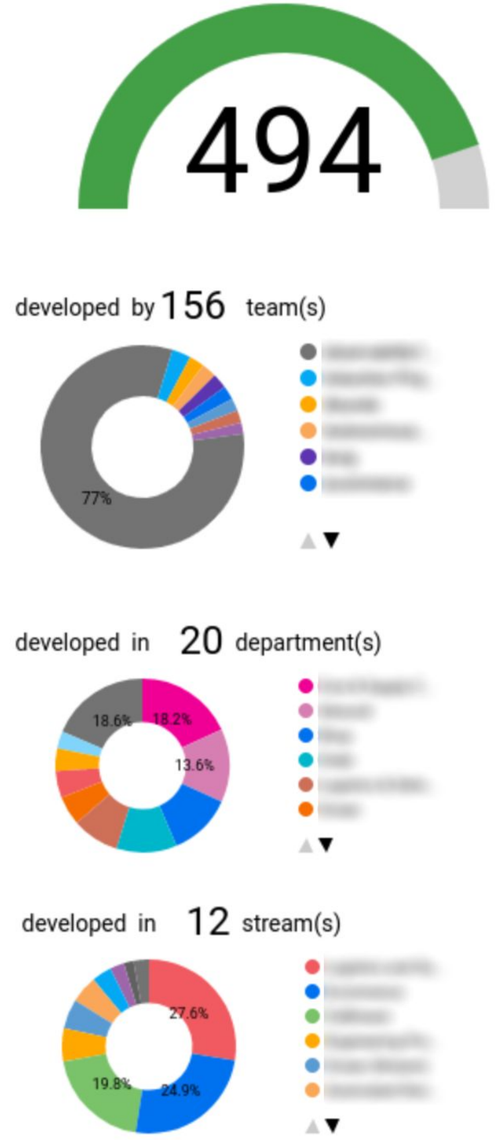
- **Blue line**: base feature active by default
  - Increasing trend due to new adopters
- **Pink line**: deprecated feature
  - Historically active by default, made opt-in
- **Purple line**: new feature
  - Initially experimental opt-in, promoted to default
- **Gray line** and **yellow line**: deprecated feature superseded by new feature



© Copyright 2023 Ocado Group plc. All rights reserved.

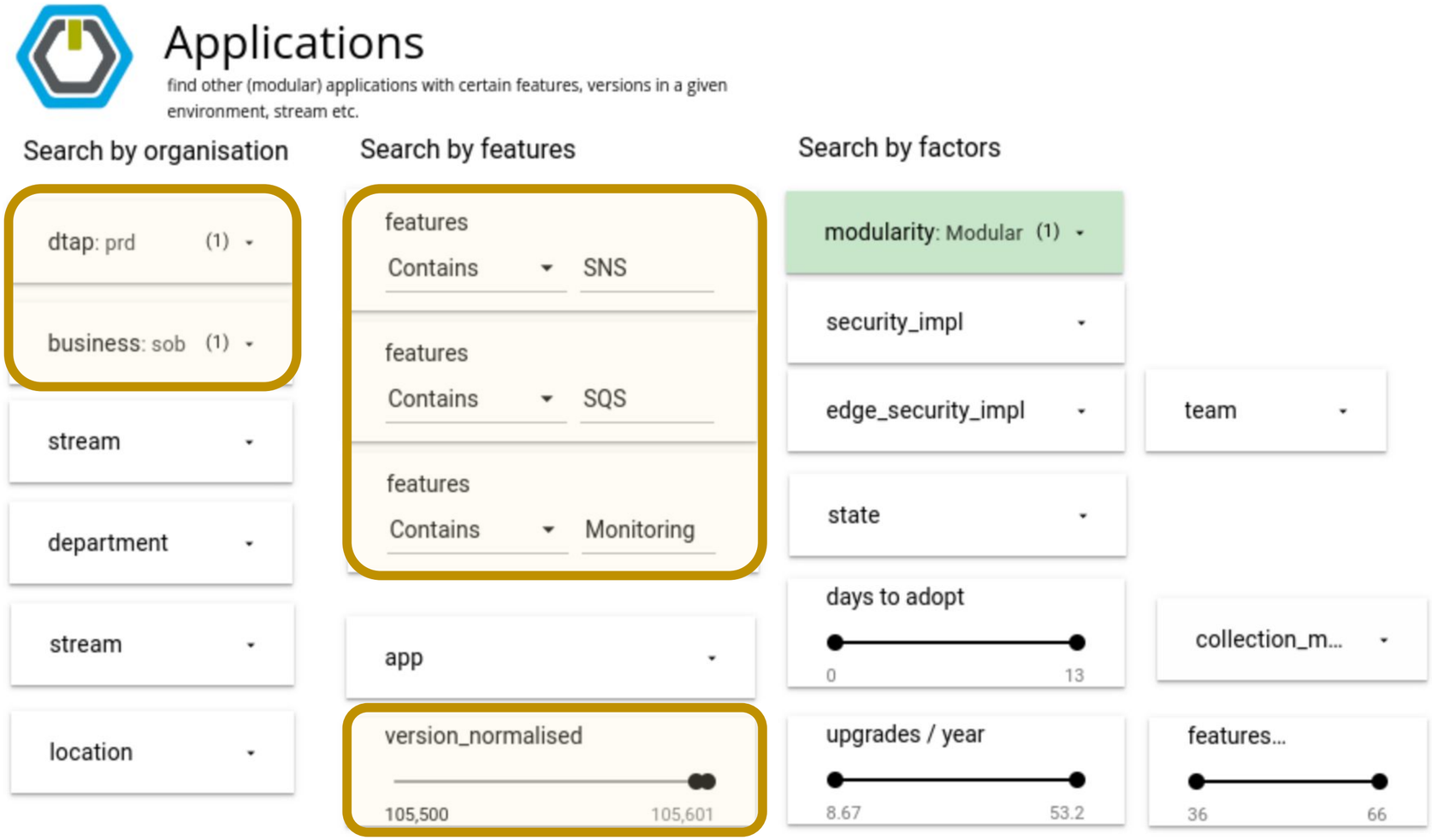
2. Example: users discovery



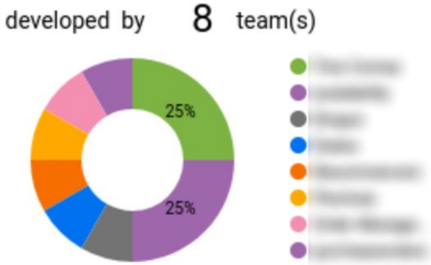


© Copyright 2023 Ocado Group plc. All rights reserved.

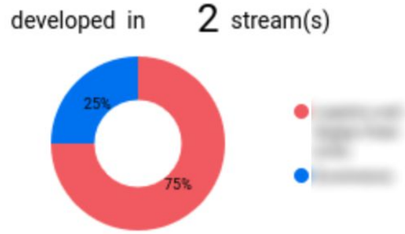
2. Example: users discovery











© Copyright 2023 Ocado Group plc. All rights reserved.

3. Interacting with people

Interviews

- Feedback asked directly contacting users
  - Experience using an existing feature
  - Suggestions for improvements
  - Expectations for an upcoming feature

Early access

- Trialling of new features
  - Restricted number of users: volunteers and directly engaged
  - Real production applications in test environments
  - Reliable feedback

Job swaps

- Community developers joining owner team
  - Limited period of time
  - Bring technical knowledge to community
  - Improve self-support
  - Potential contributors

Support channel

- Support requests, feature requests, bug reporting
  - Public Slack channel
  - 650+ users, ~25 threads/week, fast response
  - Useful insights from any request

© Copyright 2023 Ocado Group plc. All rights reserved.

4. Creating narration

News channel (~1 post/week)

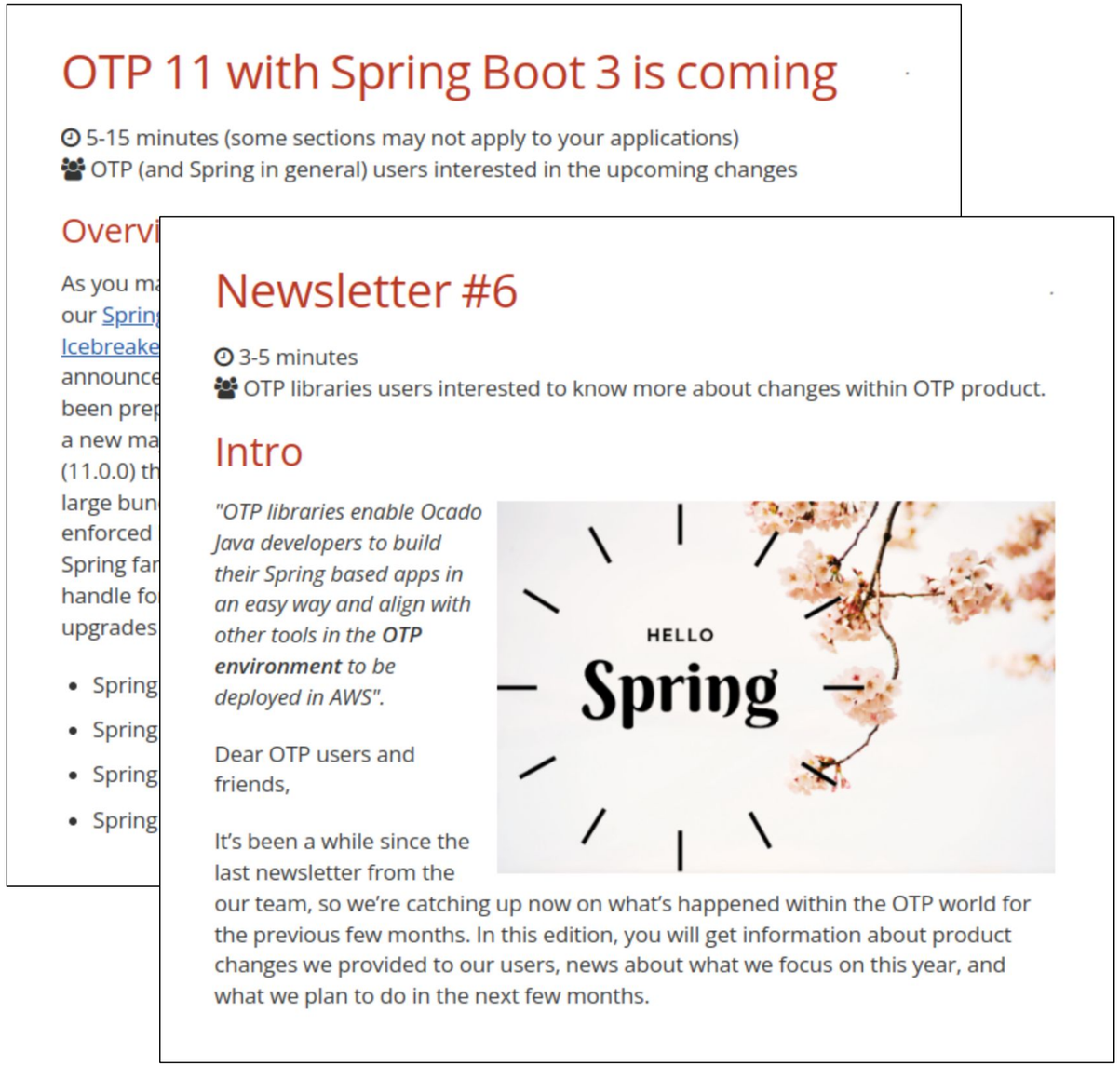
- General announcements
  - E.g.: new important release, discovered vulnerabilities, feature deprecation

Blog posts (~1 post/month)

- Important announcements
  - E.g.: new major versions, new important features, major changes

Newsletter (~1 post/semester)

- Summary of the achievements, ongoing work and upcoming changes and features



© Copyright 2023 Ocado Group plc. All rights reserved.

Working with the community

Takeaways

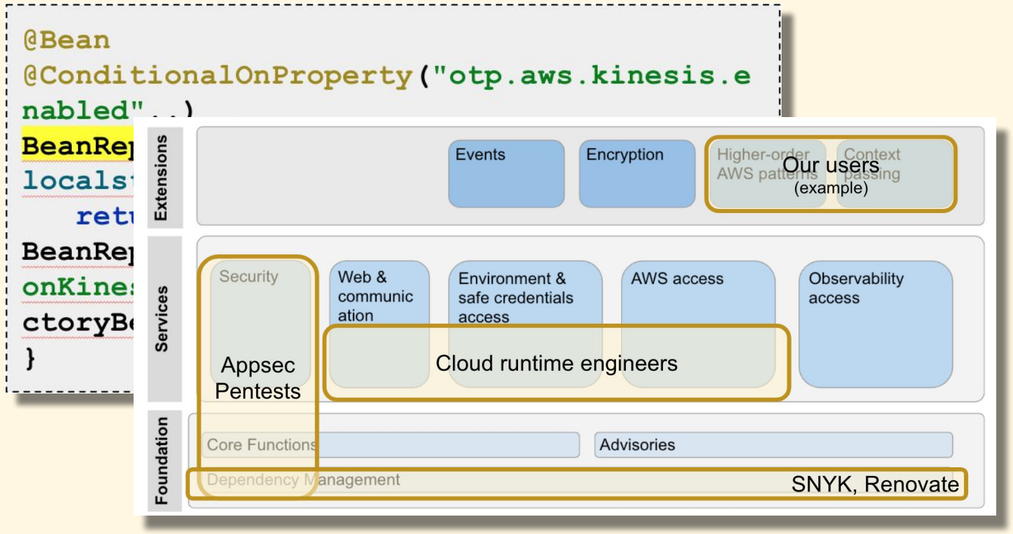
- Collect **data** to assess **reach** and impact
- Be the **custodian** but do involve **community**
- **Narrate** and **discuss** in the right proportion



© Copyright 2023 Ocado Group plc. All rights reserved.

Agenda

- About Ocado Technology
- About our libraries
- How we make it work <u>at scale</u>
  - Working with the community
  - **Software design**
  - Engineering practices



© Copyright 2023 Ocado Group plc. All rights reserved.

Software design

Our design findings

- **Domain** (high-level)
- **Modules** (low-level)

© Copyright 2023 Ocado Group plc. All rights reserved.

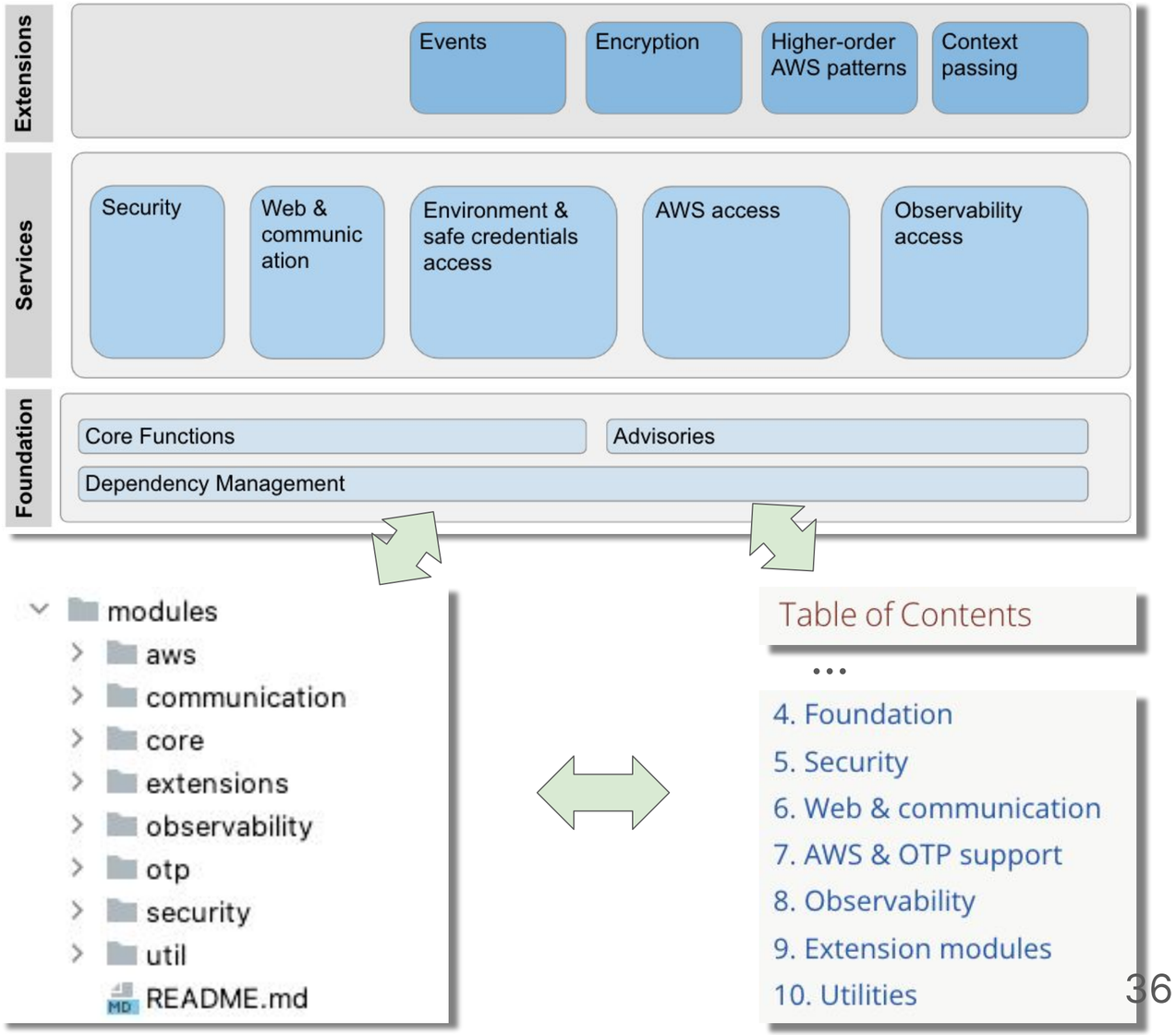
Domain design - overview

We keep (**iterative**):

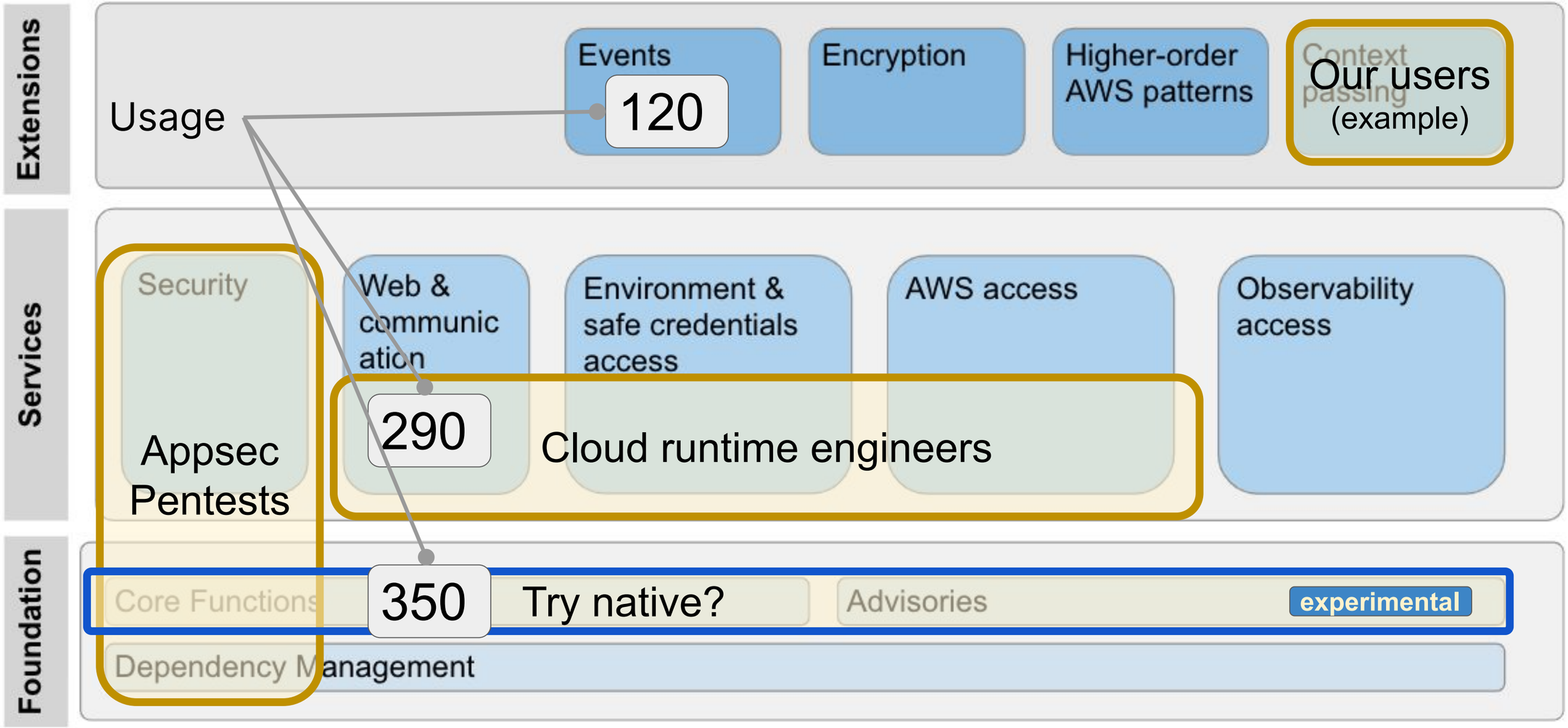
- refining modules into **layers** and **domains**
- translates to structure of:
  - repository
  - documentation

=

- Easier **collaboration** and **contribution**
- Easier **evolution**



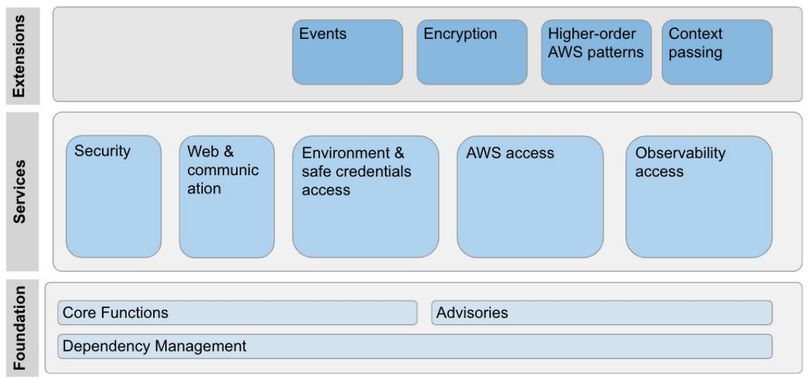
Domain design - collaboration



© Copyright 2023 Ocado Group plc. All rights reserved.

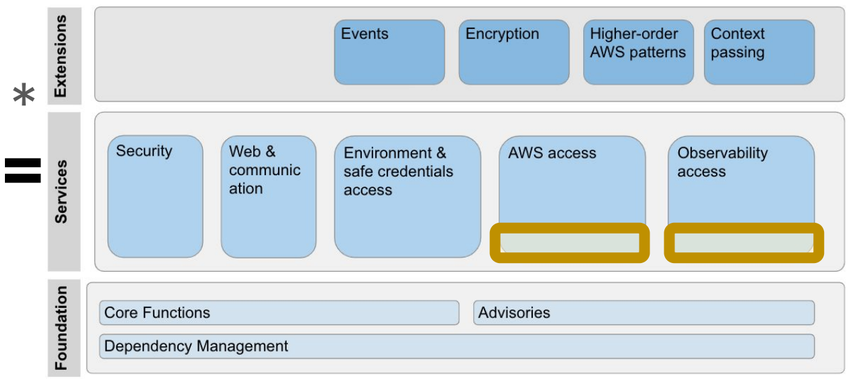
Domain design - evolution

User application



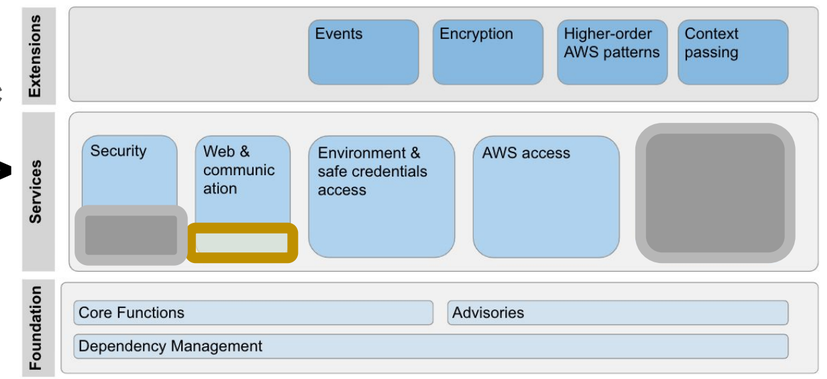
Old cloud platform [BEANSTALK]

User application
(**almost lift & shift**)



New cloud platform [ECS]
(a.o. different credentials, logging)

User application
(**pick fewer modules***)



Mesh [EKS]
(more functionality out-of-the-box)

*Floating and sinking platforms [G. Hohpe]

© Copyright 2023 Ocado Group plc. All rights reserved.

Module design

6 considerations

- Empirical, based on observations
- Help us achieve good characteristics

© Copyright 2023 Ocado Group plc. All rights reserved.

Module design

1 of 6: Do one thing, which is easy to explain

modular

simple

self-support

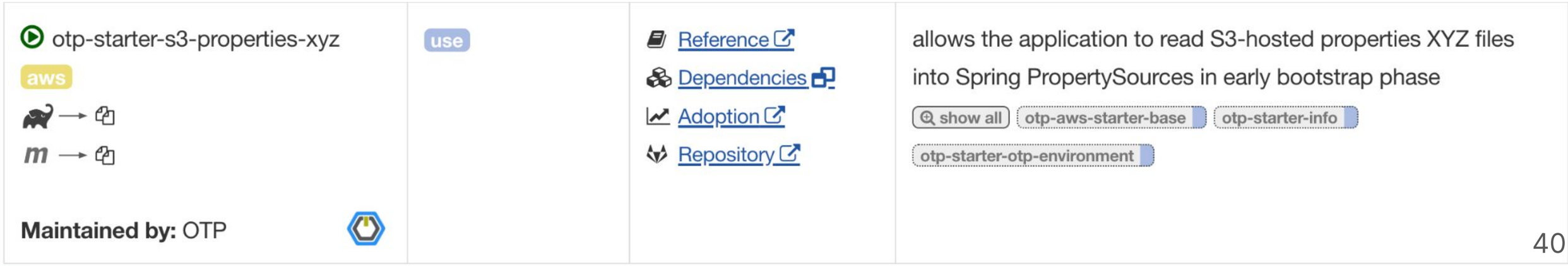
contribution

Control questions:

- Can I explain my module in a single sentence?
- Is there a separate use case for it?

Example

- *“Load a property file from known S3 location and refresh it periodically”*



Module design

2 of 6: Provide predictable @ConfigurationProperties for each module

Control questions:

- Are properties in a predictable namespace?
- Do they fit the “big picture”?
- Can I expand property tree without a breaking change?

configurable

predictable

self-support

```
otp:
 aws:
   s3:
     enabled: true
     refresh:
       enabled: true
       interval-ms:.
```

```
otp:
 aws:
   s3:
     enabled: true
     refresh: true
#can’t easily
 add new properties
```

Module design



predictable

self-support

3 of 6: Allow extensions but make clear what is API

Control questions:

- Do I want to allow users to extend behaviour?
- Do I accept coupling created this way?

Approach:

- Allow extension (@ConditionalOnMissingBean)

Consideration

- Make it clear which ones are the @Api

```
@Bean
@ConditionalOnMissingBean(..)
CertainBean certainBean(..) {
  ..
}


@Bean
@ConditionalOnMissingBean(..)
EndorsedApiBean endorsedApiBean(..){
   ..
}
```

Module design

predictable

self-support

+

testable

4 of 6: Make modules similar and natural in use

Control questions:

- Will this module behave similarly as another one?

Example:

- Our modules have:
  - on/off properties for all features
  - good loggers (+package isolation)
  - extensibility at similar level
  - similar metrics

© Copyright 2023 Ocado Group plc. All rights reserved.

Module design

5 of 6: @Conditional logic can be seen as “magic”

Control questions when working with @Conditions:

- Are **most users**:
  - .. expecting a hidden synergy?
  - .. complaining about “magic”?
  - .. in control of the @Condition?
    - (or can it trigger out-of-their-control easily)

© Copyright 2023 Ocado Group plc. All rights reserved.

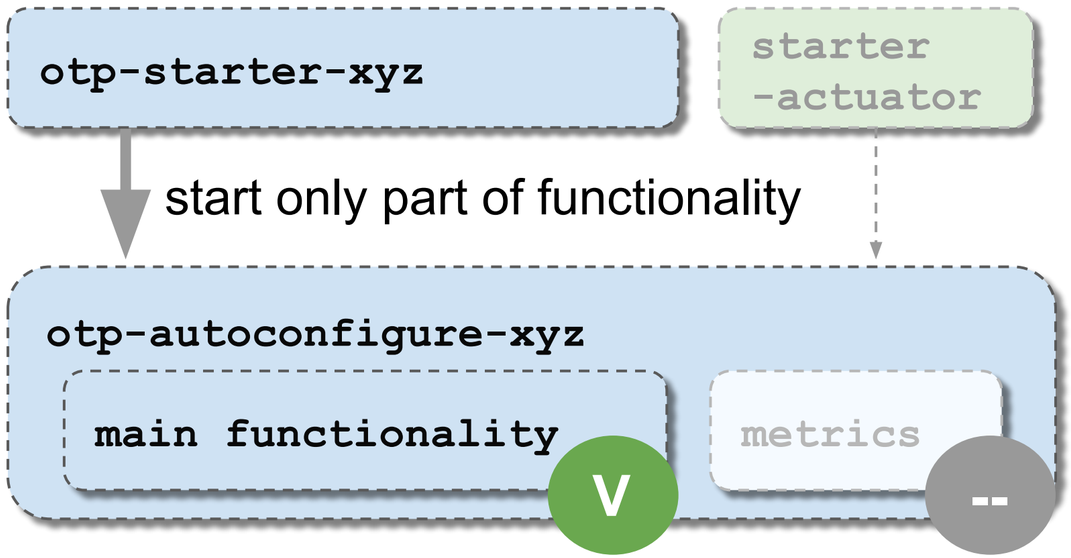
Module design

~~surprising~~

predictable

5 of 6: @Conditional logic can be seen as “magic”

Case study



- Should adding Actuator turn on metrics in my module?
  - *We assume yes (Spring Boot common sense)*
  - *.. but may still interview some users to validate*

© Copyright 2023 Ocado Group plc. All rights reserved.

Module design

modular

predictable

+

testable

6 of 6: Provide drop-in test modules

Control questions:

- When do I want to provide a test module / fake?
- Is it something that users will have to mock often?*

How:

- “Just drop-in and use” (`test` scope)
- Replaces an integration (usually)

Technical detail:

- BeanRegistryPostProcessor

```
@Bean
BeanReplacer<AnIntegration> mockIntegration()
{
    return BeanReplacer
      .<AnIntegration>replace("anIntegration")
      .byFactoryBean(AnIntegration.class);
}
```

(syntax sugar for BeanRegistryPostProcessor)

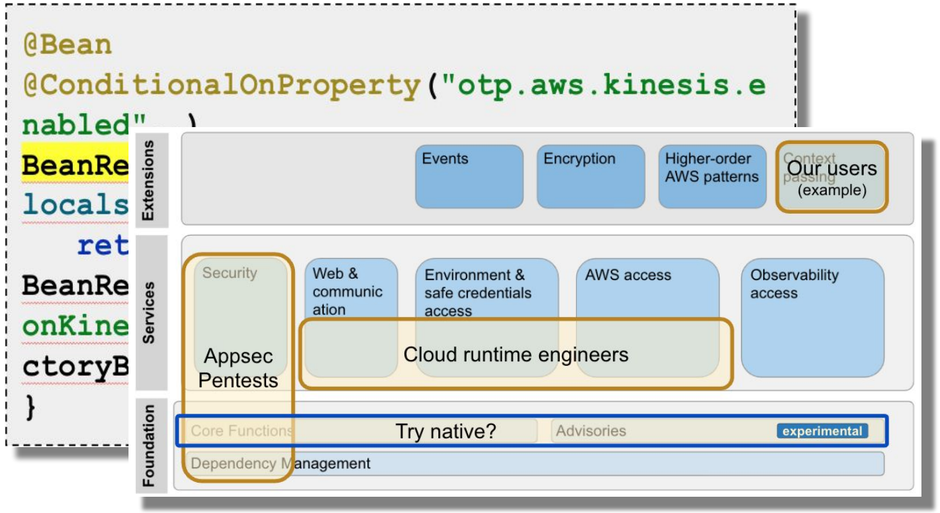
*[GoogleEng-Faking]

© Copyright 2023 Ocado Group plc. All rights reserved.

Software design

Takeaways

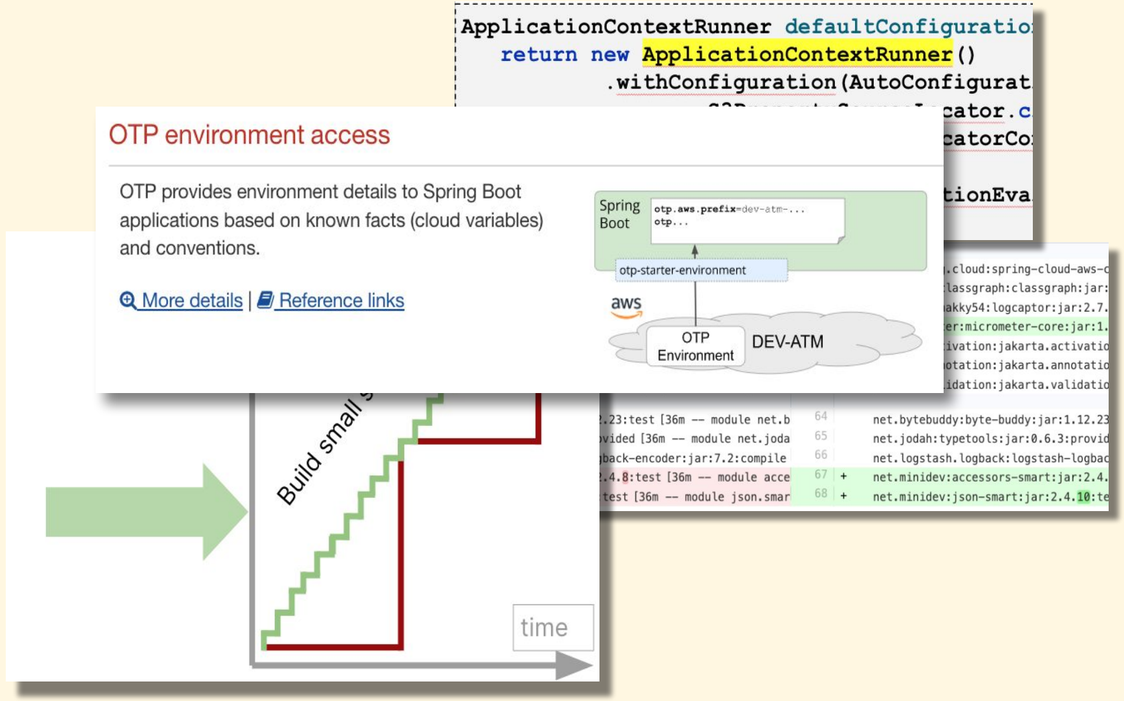
- **Domain**
  - Understand scope
  - Visualise → collaborate and evolve
- **Modules**
  - Mini-products with own lifecycle → agility
  - Simple, predictable, easy to contribute



© Copyright 2023 Ocado Group plc. All rights reserved.

Agenda

- About Ocado Technology
- About our libraries
- How we make it work at scale
  - Working with the community
  - Software design
  - **Engineering practices**



© Copyright 2023 Ocado Group plc. All rights reserved.

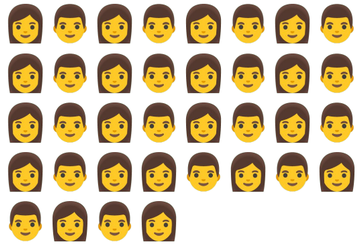
Engineering practices

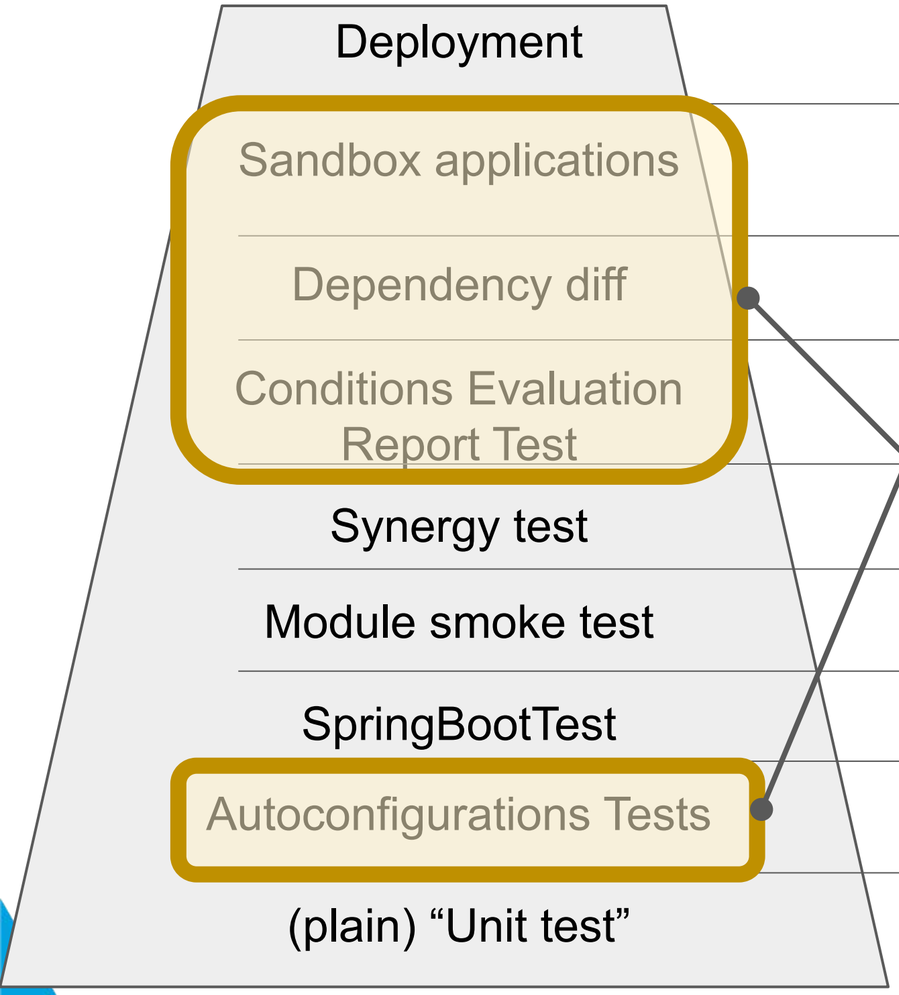
- Libraries **Testing**
- Agile **Release** model
- Agile **Documentation**

© Copyright 2023 Ocado Group plc. All rights reserved.

Libraries testing

+ Community Early Access Testing



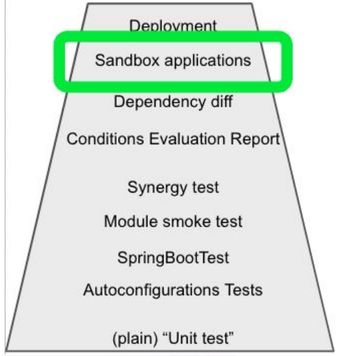


Few / slow	System integration	Deployment to cloud
	High-level regression	Spring Boot applications / Testcontainers / LocalStack
	Classpath regression	diff2html-cli
	High-level autoconfig regression	Spring Boot Test / Conditions Evaluation Report
	Module combinations	Spring Boot applications
	Integrity control	Spring Boot Test
	Configuration correctness	Spring Boot Test
	Module-level autoconfig regression	ApplicationContextRunner
Many / fast	Logic	Unit Tests / mocking frameworks

© Copyright 2023 Ocado Group plc. All rights reserved.

Libraries testing

1 of 4: Sandbox



Use case:

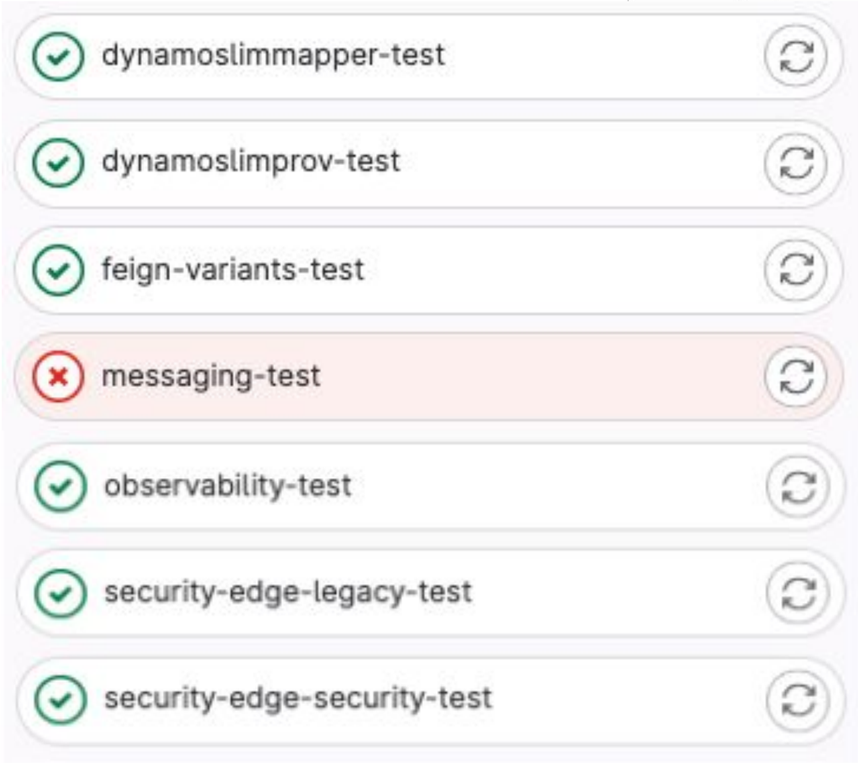
- High-level regressions

Setup:

- 12 “combo” applications (cover many functions)
- Mocked neighbourhood, run in parallel

Technical:

- Docker Compose + **LocalStack** (AWS)
- Testcontainers



• • •

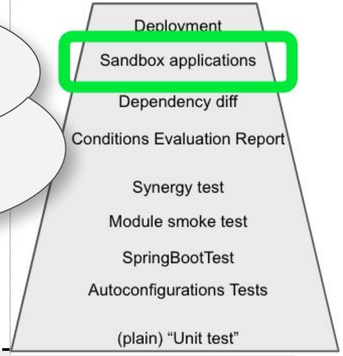
© Copyright 2023 Ocado Group plc. All rights reserved.

Libraries testing

1 of 4: Sandbox

Good for onboarding and ad-hoc tests,

May adopt generic containers and `spring-boot:test-run` soon



test

```
@Testcontainers
public class LocalCloudAppTest extends TestContainer {

    (...)

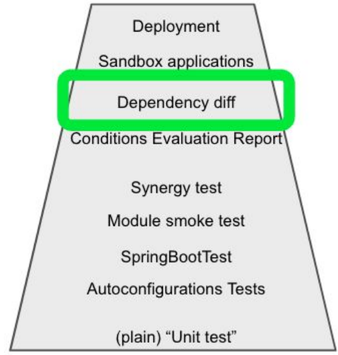
    @BeforeEach
    public void before() {
      RestAssured.enableLoggingOfRequestAndResponseIfValidationFails();
    }

    @Test
    public void given...Then...() {
        given()
                ...
                .get(applicationUrl() + "/allowedForEditor")
                .then().assertThat().statusCode(200);
    }
```

docker-compose.yml

```
services:
    app ...
    localstack ...
    authserver ...
    neighbourhood ...
```

Libraries testing



2 of 4: Dependency Diff

Use case:

- Dependencies regressions*
  - Upgrades (e.g. Spring)
  - Own rework

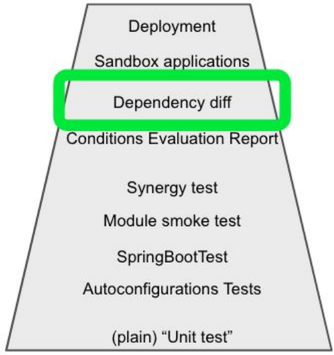
Setup:

- Transitive dependency diff (all modules)
- `diff2html-cli`
- Manual check (hard to automate)

*Our users’ classpaths

© Copyright 2023 Ocado Group plc. All rights reserved.

Libraries testing



2 of 4: Dependency Diff

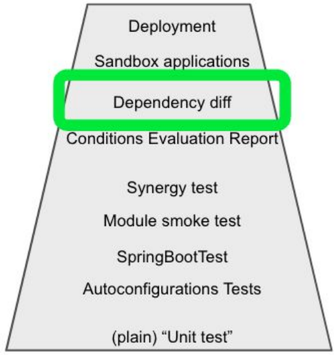


Expected change: Spring Boot 2.7.10 → 2.7.11 upgrade, change

```
@@ -1,7 +1,7 @@
-    ch.qos.logback:logback-classic:jar:11.7.10 [36m -- module logback
-    ch.qos.logback:logback-core:jar:1.2.10 [36m -- module logback.co
+    ch.qos.logback:logback-classic:jar:12.7.11 [36m -- module logback
+    ch.qos.logback:logback-core:jar:1.2.11 [36m -- module logback.co
```

© Copyright 2023 Ocado Group plc. All rights reserved.

Libraries testing



2 of 4: Dependency Diff



Unwanted change: scope change harmful for users (@Conditions)

```
66      org.slf4j:jul-to-slf4j:jar:1.7.36:test [36m -- module jul.to.slf4j [0;
67      org.slf4j:slf4j-api:jar:1.7.36:compile [36m -- module org.slf4j [0;1m
68  -   org.springframework.boot:spring-boot-autoconfigure:jar:2.7.10:compile
69  -   org.springframework.boot:spring
70  -   org.springframework.boot:spring
```

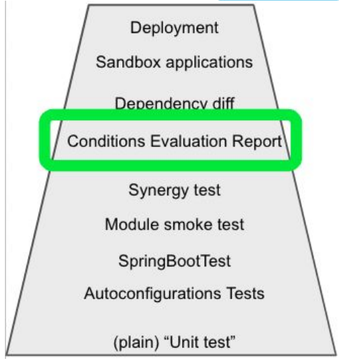
2.7.10:test

```
66      org.slf4j:jul-to-slf4j:jar:1.7.36:test [36m -- module jul.to.slf4j [0;
67      org.slf4j:slf4j-api:jar:1.7.36:compile [36m -- module org.slf4j [0;1m
68  +   org.springframework.boot:spring-boot-autoconfigure:jar:2.7.11:compile
69  +   org.springframework.boo
70  +   org.springframework.boo
```

2.7.11:compile

© Copyright 2023 Ocado Group plc. All rights reserved.

Libraries testing



3 of 4: Conditions Evaluation Report Test

Use case:

- Autoconfiguration regressions
  - Own code
  - (Detect) new and interesting Spring autoconfigurations

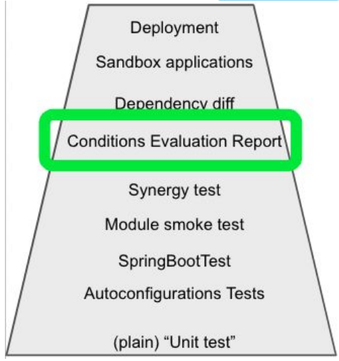
Setup:

- Unit Tests
- `assert` Spring Boot’s Conditions Evaluation Report

© Copyright 2023 Ocado Group plc. All rights reserved.

Libraries testing

3 of 4: Conditions Evaluation Report Test



expectat

```
conditions:
   positive:
        - AdvisoriesV1Au
        - AdvisoriesV1Au
        - TaskScheduling
        - TaskScheduling
        - TaskScheduling
        (...)
   unconditional: ...
```

test

```
@SpringBootTest(...)
@EnableConfigurationProperties(Conditions.class)
public class ...{

    @Autowired
    ConfigurableApplicationContext context;

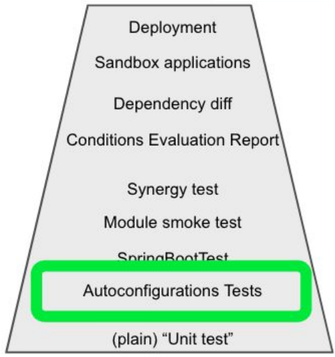
    @Autowired
    Conditions conditions;

    @Test
    public void positiveMatchesEvaluationReport() {
        assertThat(conditionReportPositiveMatches(context))
                .containsExactlyInAnyOrder(conditions.getPositive());
    }
```

© Copyright 2023 Ocado Group plc. All rights reserved.

Libraries testing

4 of 4: Autoconfiguration variants



Use cases:

- Complex @Conditions coverage in autoconfigurations
  - Classpath and property variants

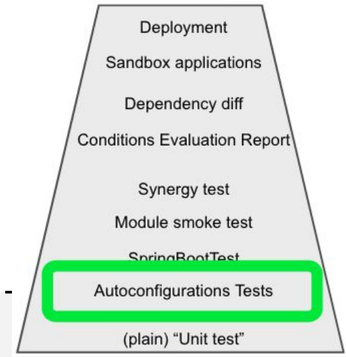
Setup:

- Spring Boot’s ApplicationContextRunner

© Copyright 2023 Ocado Group plc. All rights reserved.

Libraries testing

4 of 4: Autoconfiguration variants



test

```
ApplicationContextRunner defaultConfiguration() {
    return new ApplicationContextRunner()
            .withConfiguration(AutoConfigurations.of(
                    S3PropertySourceLocatorConfiguration.class
            ))
            .withInitializer(...);
}

@Test
void whenS3FeatureDisabled_dontRegisterPropertySourceAndFlag() {
    defaultConfiguration()
            .withBean(AmazonS3Client.class)
            .withPropertyValues("otp.aws.s3.enabled=false")
            .run(
                    context -> assertThat(context)
                            .doesNotHaveBean(S3PropertySourceLocator.class)
                            .doesNotHaveBean("S3PropertySourceLocatorFeature")
            );
}
```

Pros:

- very fast compared to Boot tests

Cons:

- autoconfig selection & classpath filtering requires care

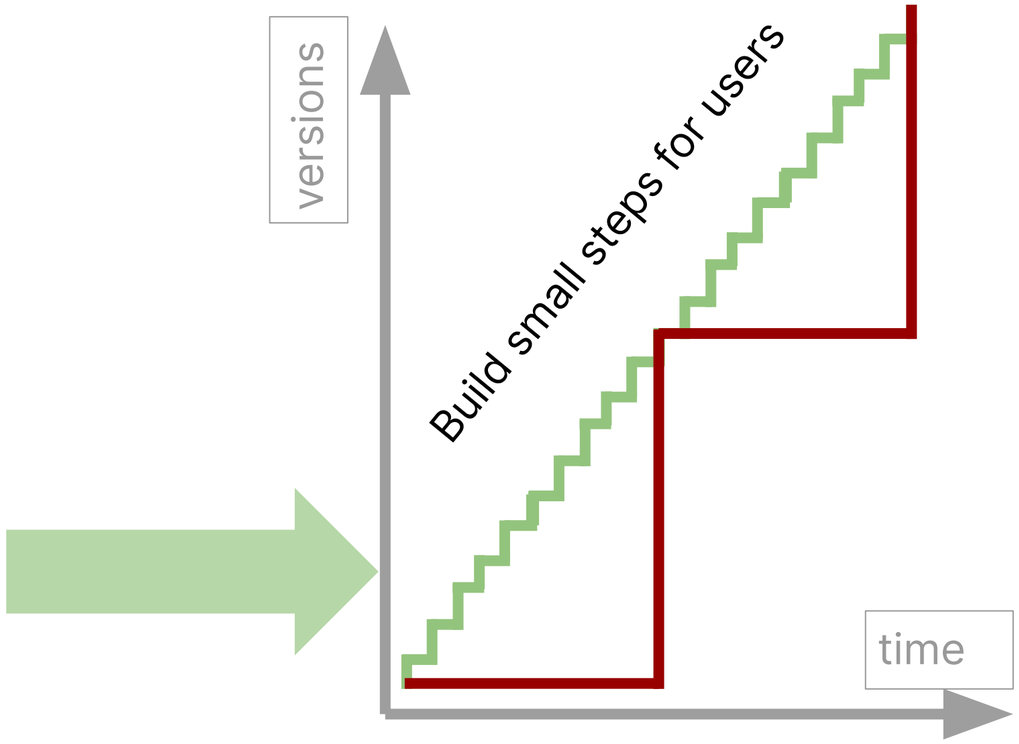
Agile release model

The challenge

- Produce 200+ releases a year ..
- But what about big releases (e.g. Spring Cloud ILFORD, Spring Boot 3.0, Java 17, etc.?)

Solution

- Learn what's coming up (months ahead)
- Early milestones (SNAPSHOT, RC) with community
- Release as much as possible in advance
- Mini-versions + advertise agile adoption



Tactics

- Empathy!
- Use usage / adoption data for every decision
- Enough for a separate presentation :)

Agile release model

Benefits

- One version,
- Up-to-date (no fear of 0-day exploits)
- Engaging (community)

Downsides

- A lot of effort
- Many artifacts in Nexus

© Copyright 2023 Ocado Group plc. All rights reserved.

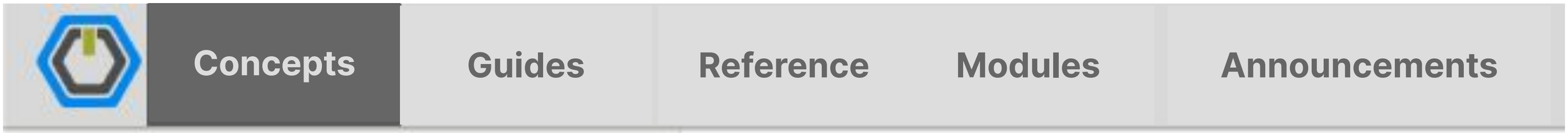
Agile documentation

The challenge

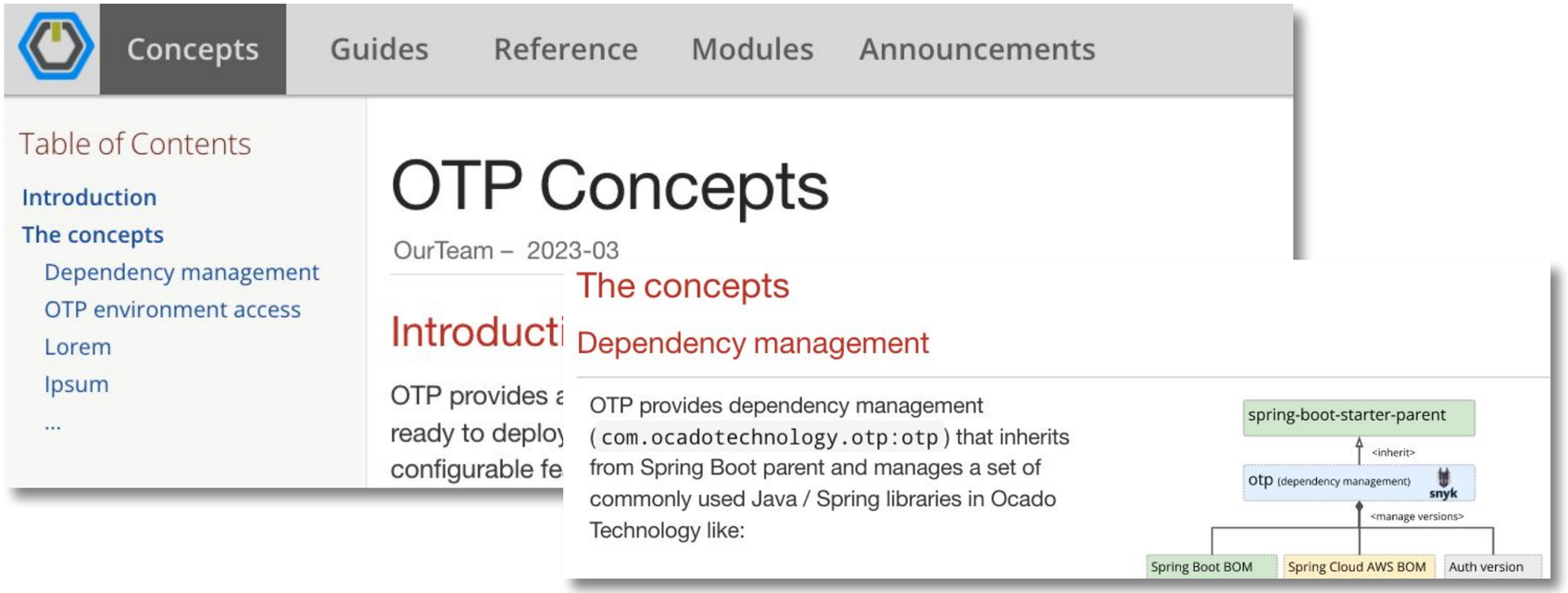
- big community → very good documentation
  - in self-support contribution and many other ways
  - It must be up-to-date but maintainable

© Copyright 2023 Ocado Group plc. All rights reserved.

Agile documentation - overview



how things work



© Copyright 2023 Ocado Group plc. All rights reserved.

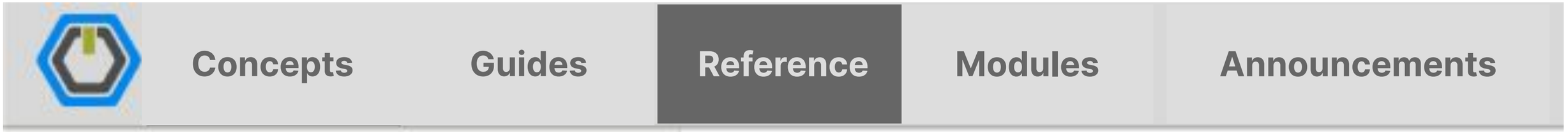
Agile documentation - overview



getting started and topical guides
secondary purpose: hands-on workshops

© Copyright 2023 Ocado Group plc. All rights reserved.

Agile documentation - overview



a one-pager (Ctrl + F)

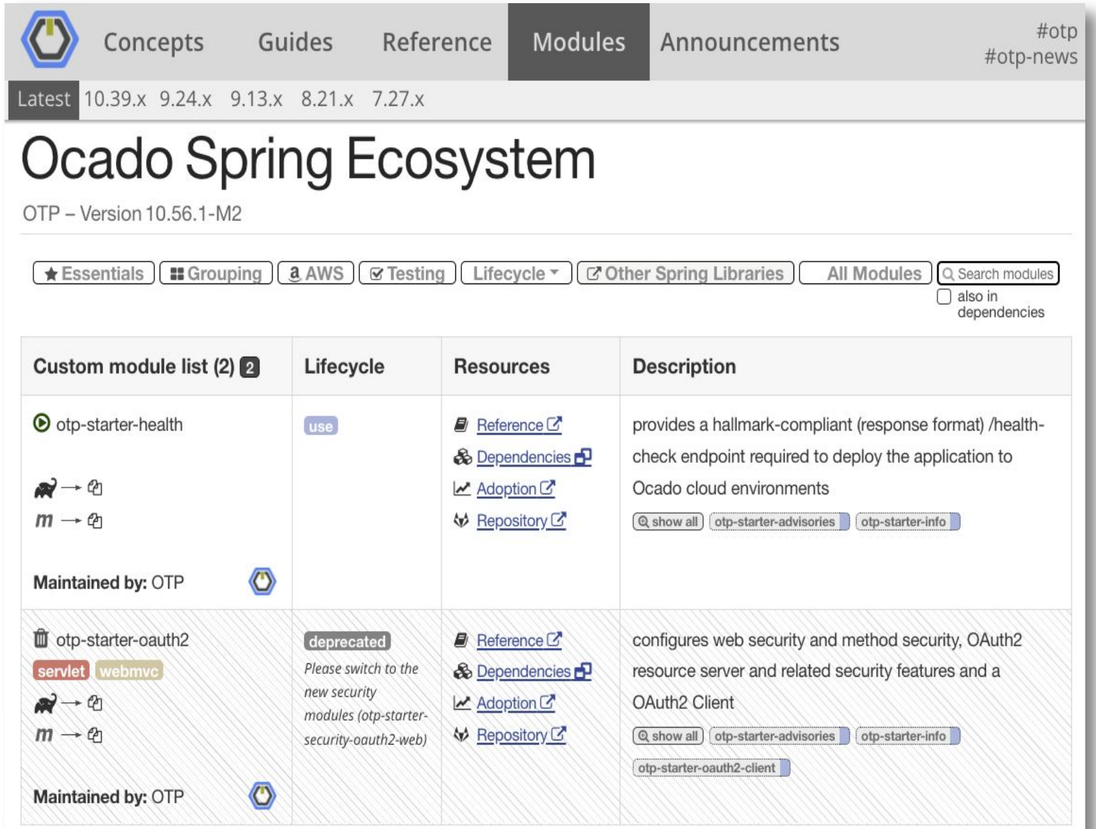
(full reference, properties, upgrade guides, change log)

© Copyright 2023 Ocado Group plc. All rights reserved.

Agile documentation - overview



module reference and search with tabs



Agile documentation - overview



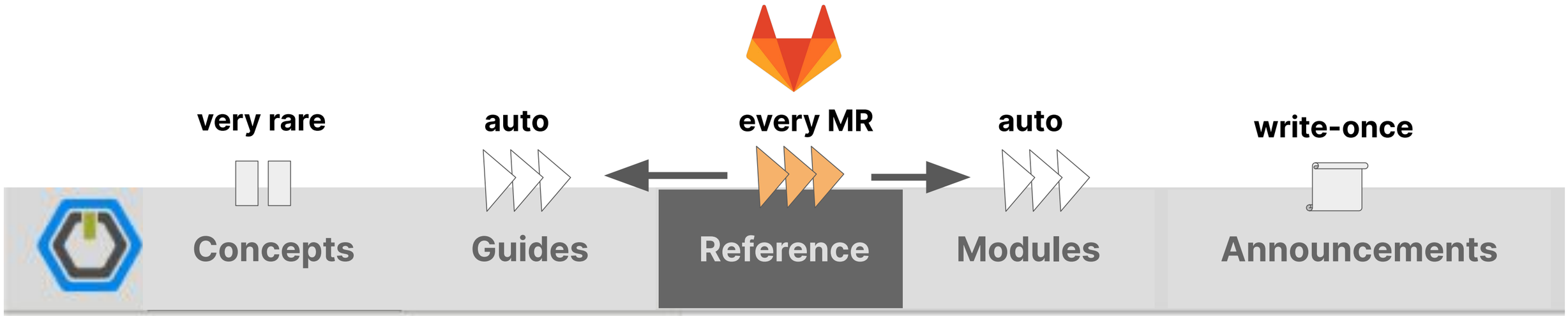
our blog

© Copyright 2023 Ocado Group plc. All rights reserved.

Agile documentation

How we lower (documentation) maintenance cost?

- **Reference -** integral part of work (every MR)
- **Some parts -** mostly auto
- **Other parts -** very rare / one-off

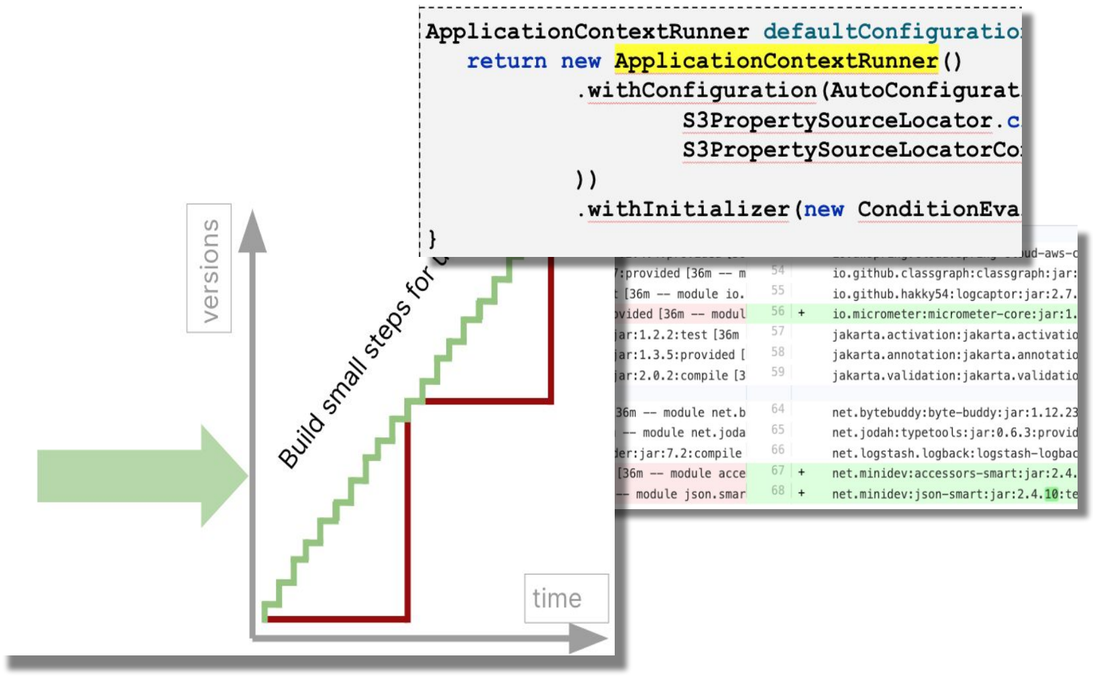


© Copyright 2023 Ocado Group plc. All rights reserved.

Engineering practices

Takeaways

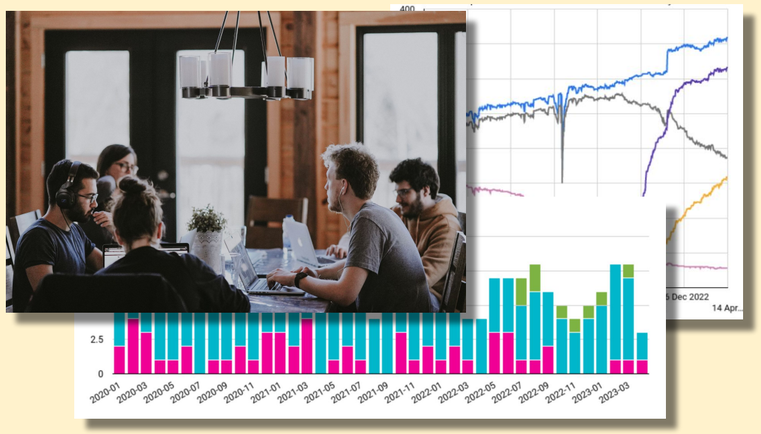
- **Libraries testing**
  - different focus → other types of tests
  - 120 teams → worth to have good tests
- **Agile release model**
  - Look ahead
  - Break waterfalls
  - Involve community
- **Documentation for large communities**
  - Multi-faceted
  - Living
  - Automated



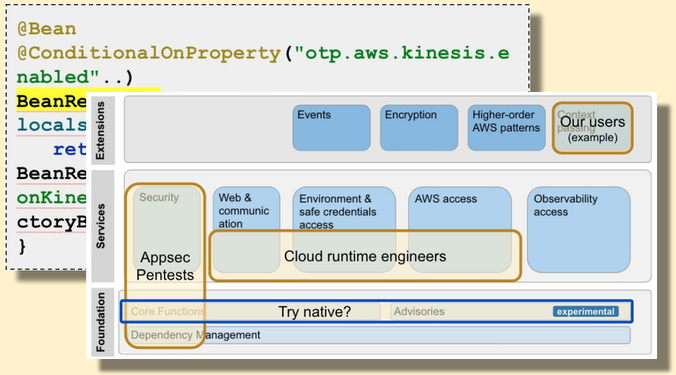
© Copyright 2023 Ocado Group plc. All rights reserved.

Summary: how we make it work at scale

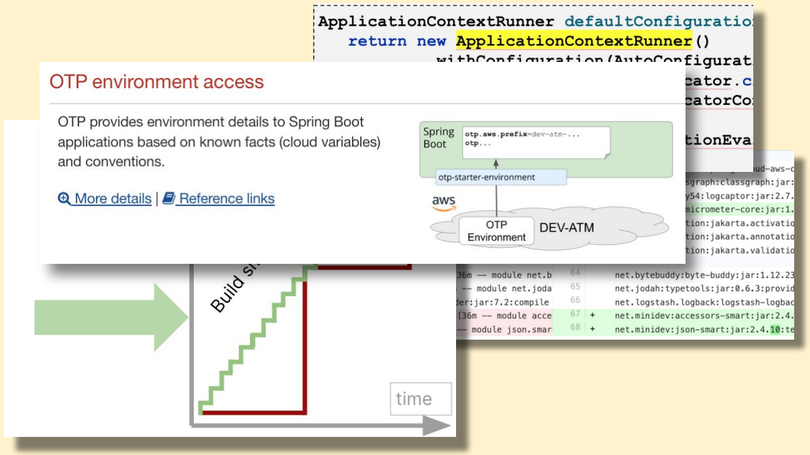
Working with the community



Software design



Engineering practices



© Copyright 2023 Ocado Group plc. All rights reserved.

Sources

- **[OcadoTech-Handbook]** https://handbook.ocado.tech/

- **Spring Boot: for library development:** Autoconfigurations, BeanPostProcessor, BeanRegistryPostProcessor, Conditions Evaluation Report, ApplicationContextRunner, @ConfigurationProperties, @ConditionalOnMissingBean
- **[Richards-Ford]** Fundamentals of Software Architecture: An Engineering Approach, Richards & Ford, O'Reilly 2020 - architecture characteristics (*..ielities*)
- **[G. Hohpe]** The magic of platforms, G. Hohpe, Platformengineering.org https://platformengineering.org/talks-library/the-magic-of-platforms
- **[GoogleEng-Faking]** Software Engineering at Google (section "Faking")
- **[PragmaticEngProd-Kanat]** [$] Pragmatic Engineer: Measuring Engineering Efficiency at LinkedIn, G. Orosz, M. Kanat-Alexander https://newsletter.pragmaticengineer.com/p/linkedin-engineering-efficiency
- **[Fowler-SharedLib]** https://martinfowler.com/articles/micro-frontends.html#SharedComponentLibraries (governance)

© Copyright 2023 Ocado Group plc. All rights reserved.

Thank you! Q & A time

Ocado Technology Barcelona

© Copyright 2023 Ocado Group plc. All rights reserved.