

Taming the Home Connect Microservice Jungle

The Backstage Software Catalog



Thomas Richer B/S/H/
Tilman Masur TNG

TNG Big Techday
2023-07-07

Who Is BSH & What Is Home Connect?

Home Appliances for Daily Use



Oven Cooking



Surface Cooking & Ventilation



Cooling



Dish Care



Laundry Care



Small Appliances



Waterbase

Home appliances under the brands



SIEMENS

GAGGENAU

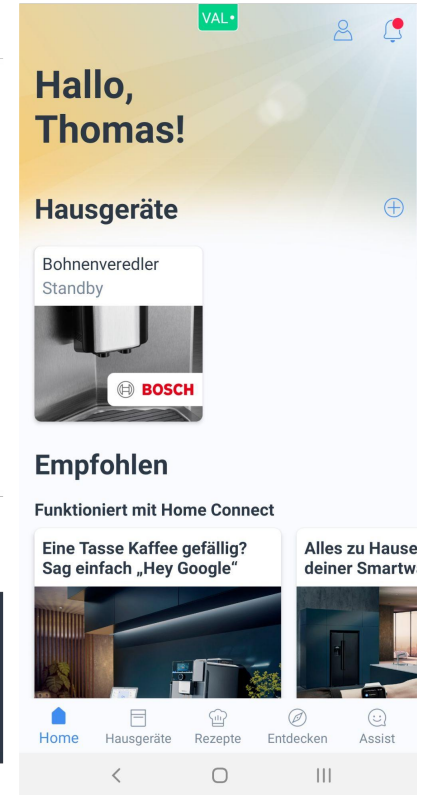


PITSOS

PROFILO

JUNKER

Ecosystem brand



Home Connect is part of the BSH Group



BSH is no. 1
in Europe



39
production sites



0
CO₂ emissions



€15.9 bn
turnover



63,000
employees



5.3%
R&D spending



Home Connect

is available
globally

58

partner integrations
live

64



countries
rolled out

18⁺

million



Home Connect enabled
appliances sold

30



languages
translated

How We Got Here: Microservices

The Monolith

- Historically default architecture
 - Whole software build and deployed as one thing
- Advantages
 - Simpler solution architecture
 - Efficient because fewer remote calls
- Downsides
 - Multiple teams working on same code base
 - Slow to deploy
 - May be badly factored
 - Can only be scaled as a unit



Microservices

- Smaller well-factored services communicating over network
- Advantages
 - Single responsible team per microservice
 - Quick to deploy
 - Forces good factoring
 - Horizontal scaling built in
 - Choose best tech per service
 - More resilient



⇒ Microservices empower teams!

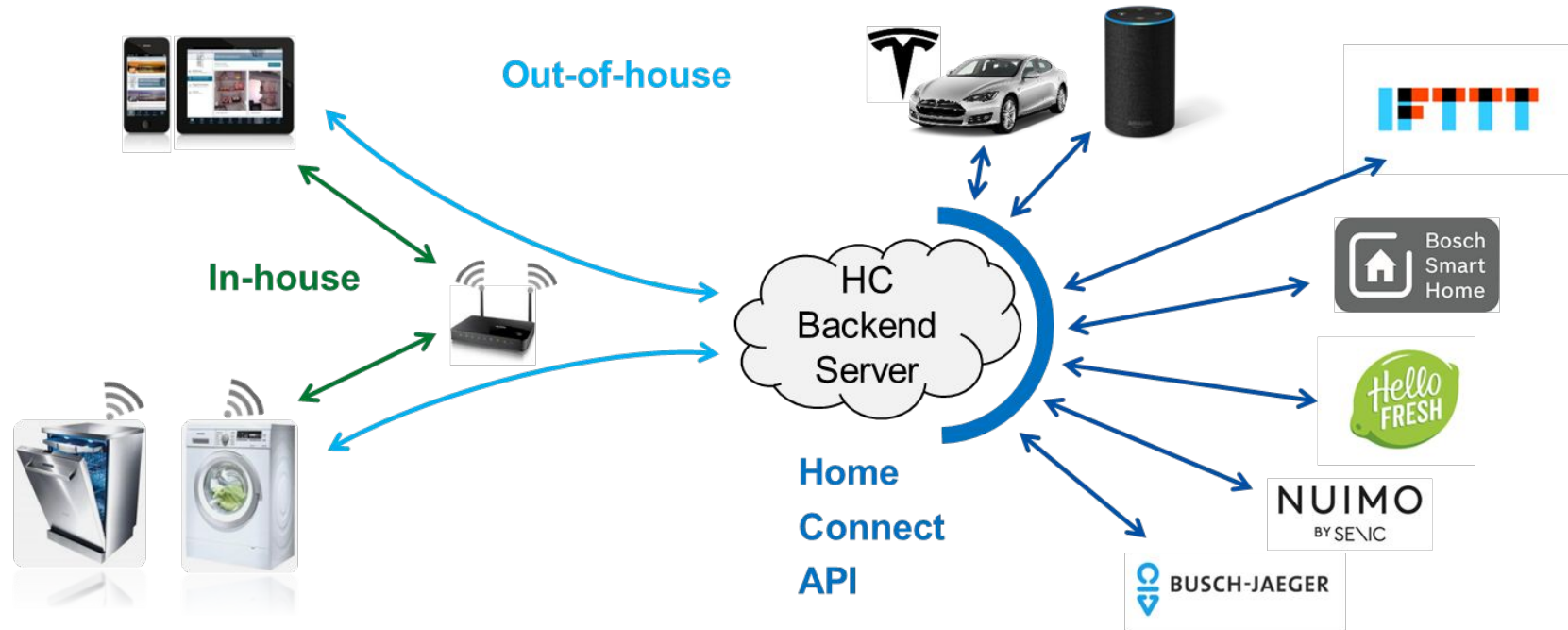
But More Autonomy Can Lead to

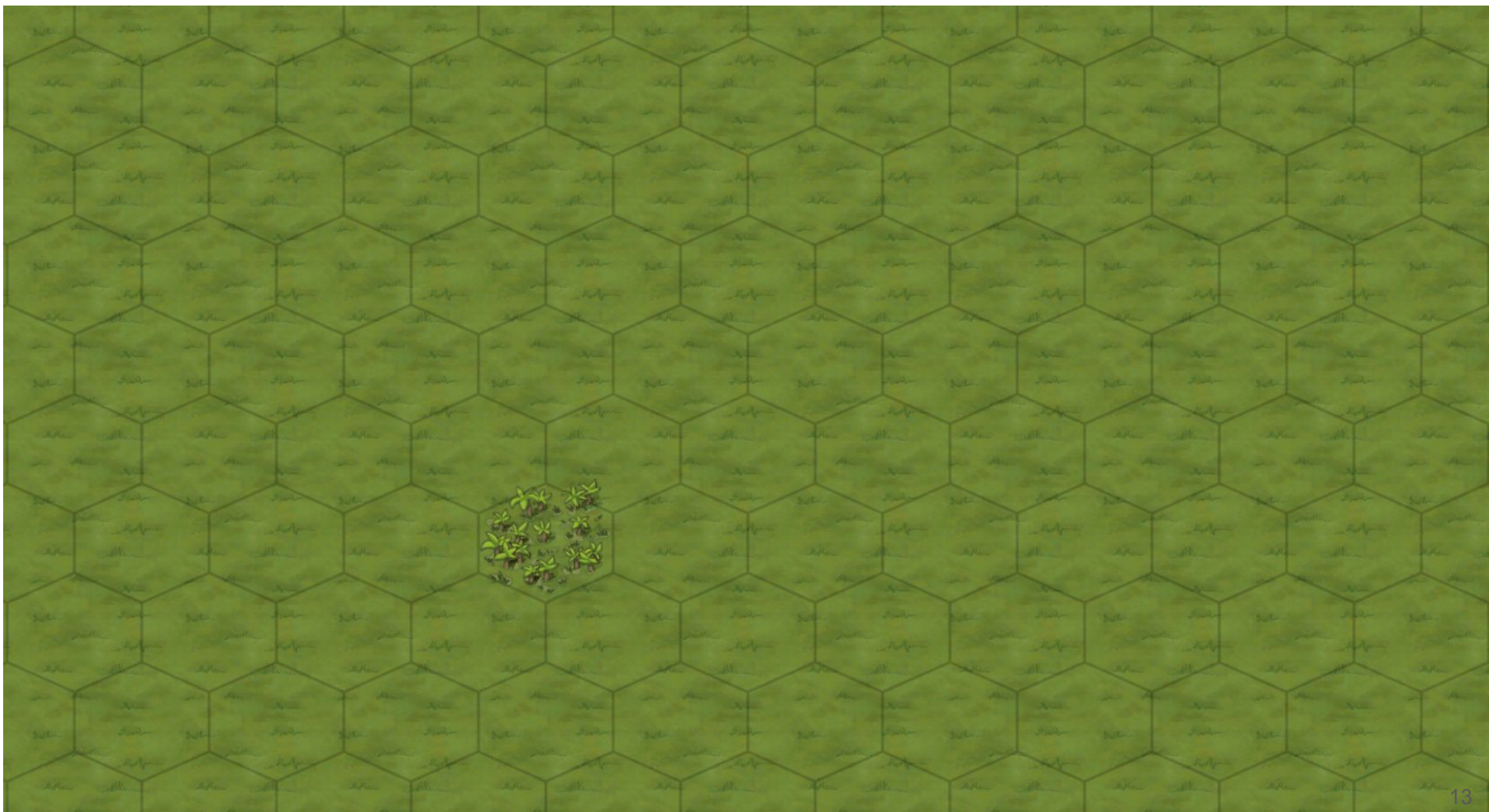
- Ecosystem fracturing faster with respect technologies, conventions, tooling
- Less reusability, knowledge sharing
- More complex architecture & org-structure
- Less overview for architects, business, & developers

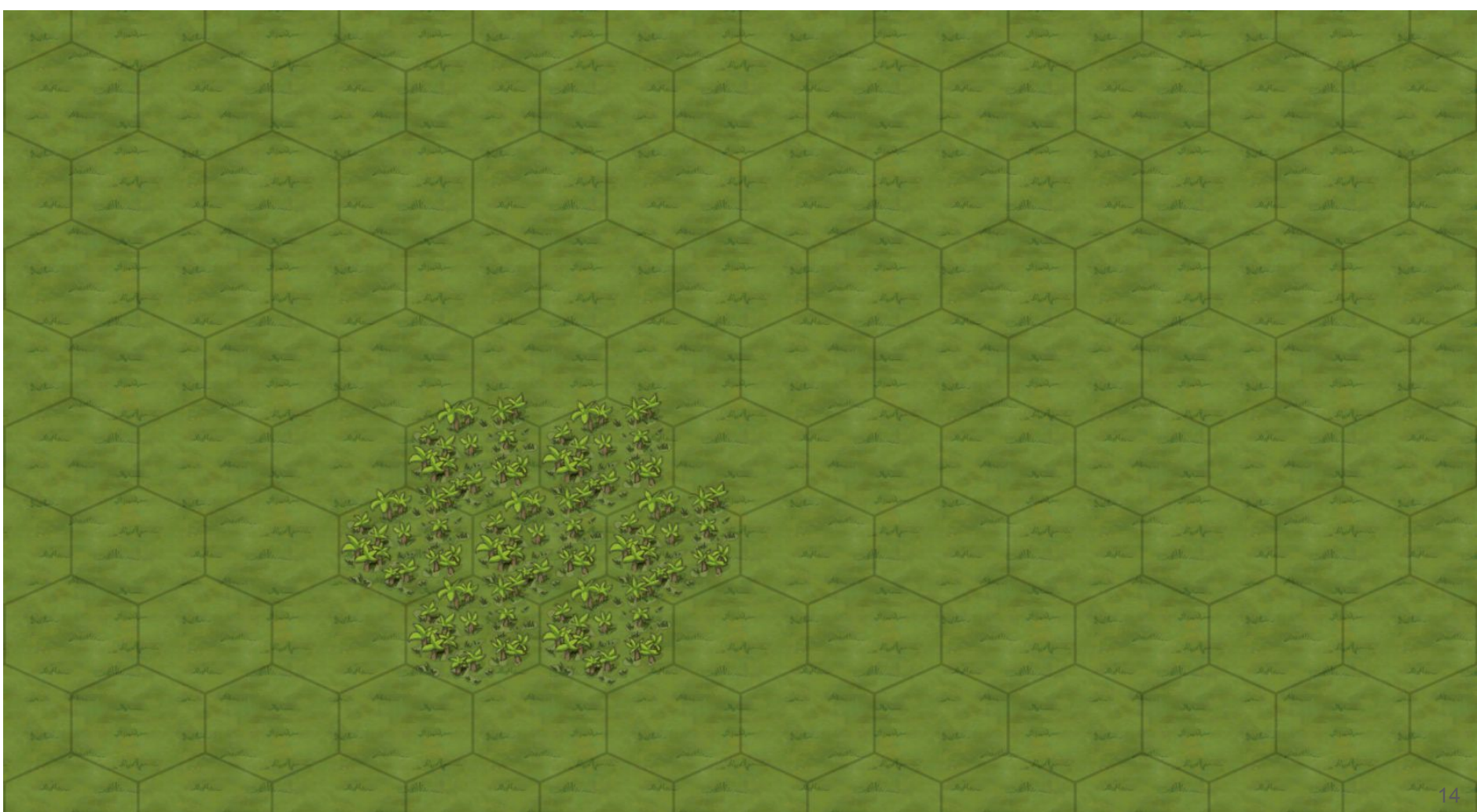


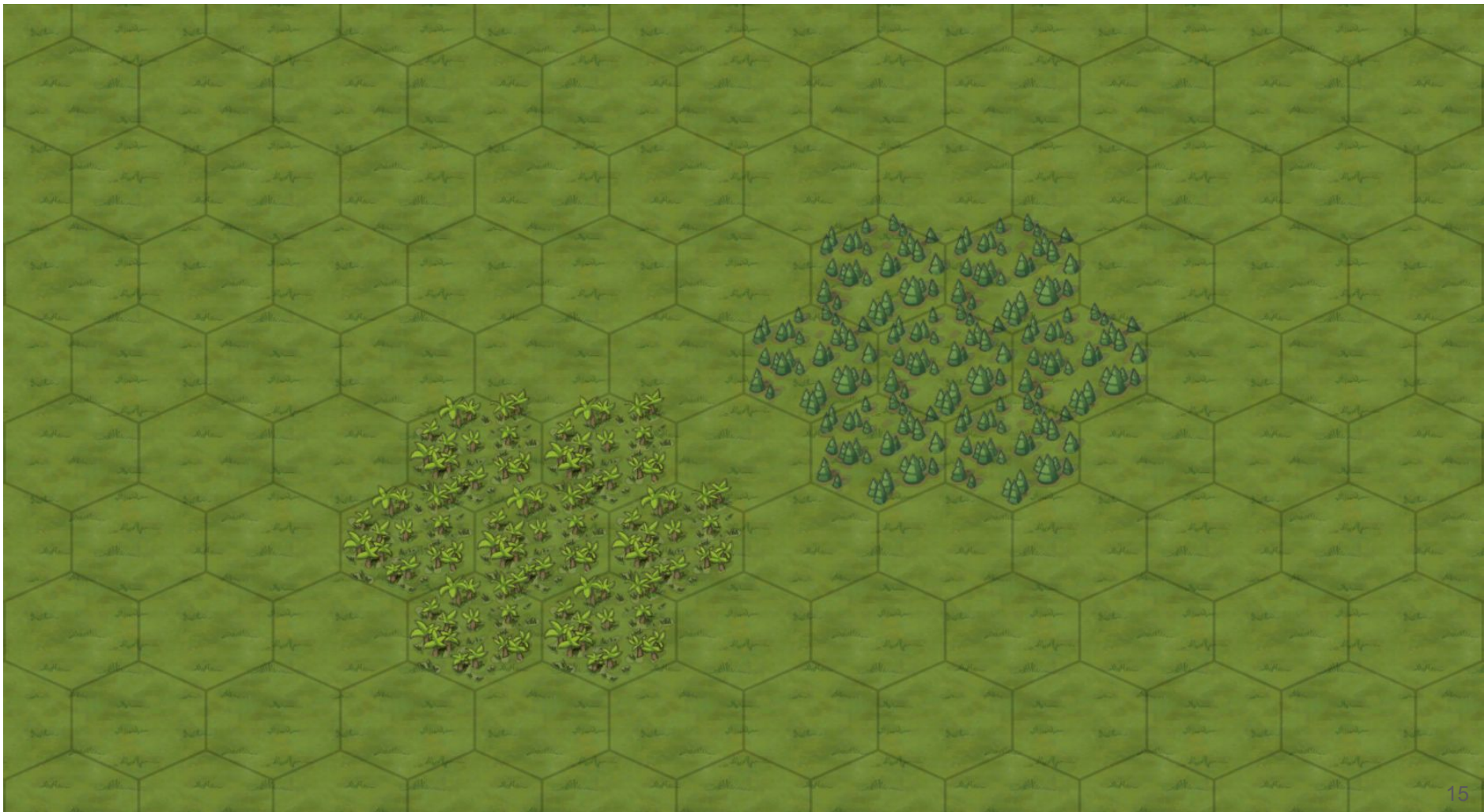
Microservices Evolution at Home Connect

Home Connect in a Nutshell









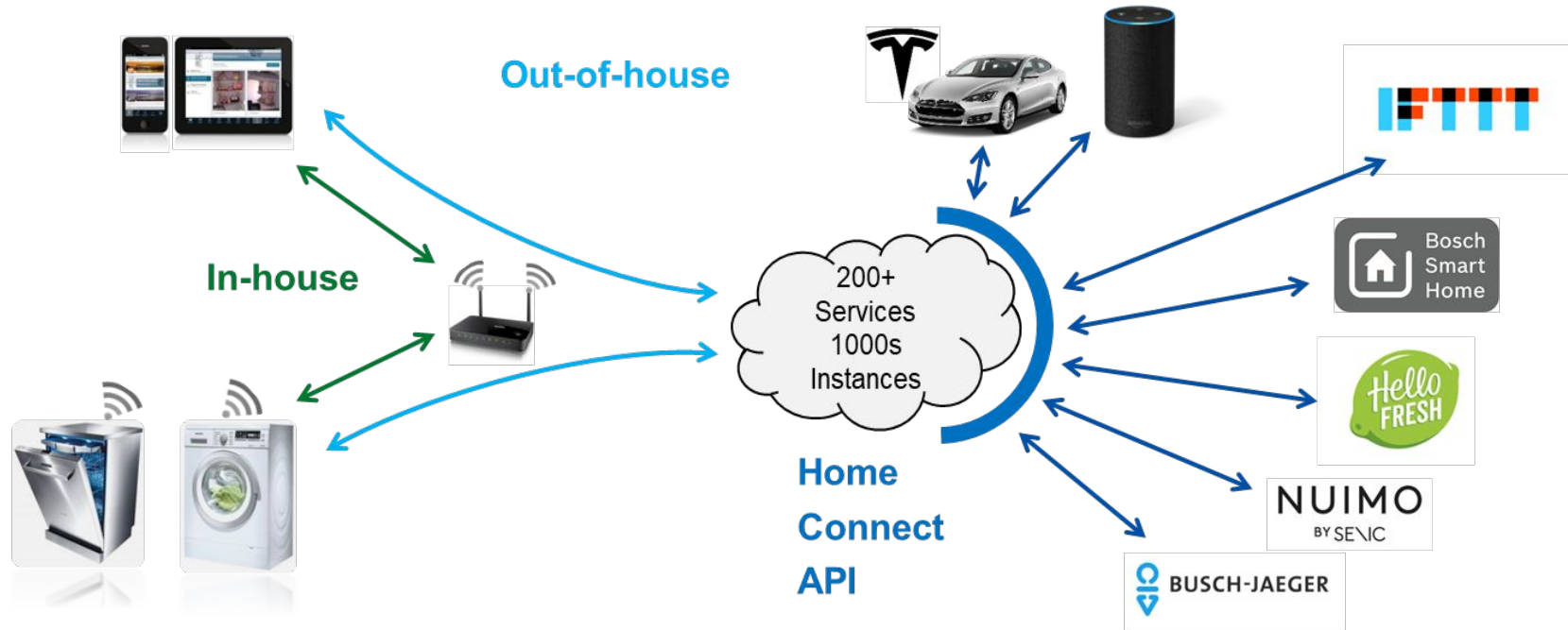


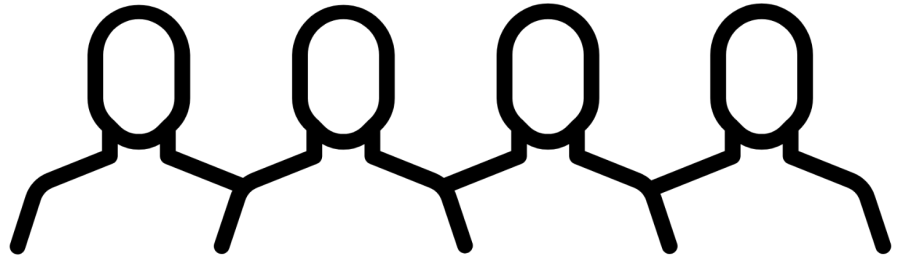
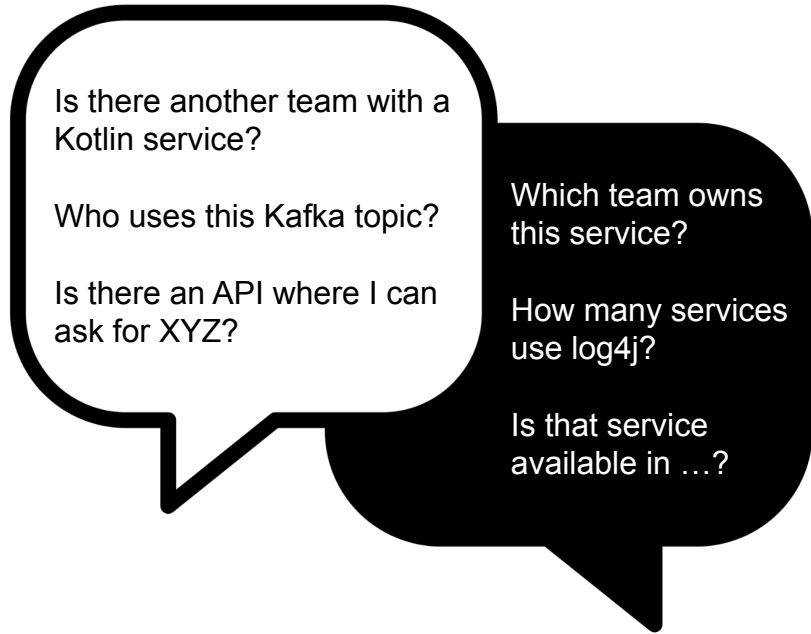
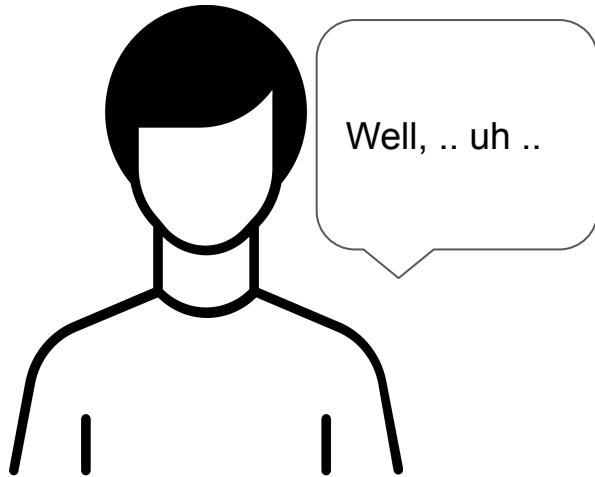






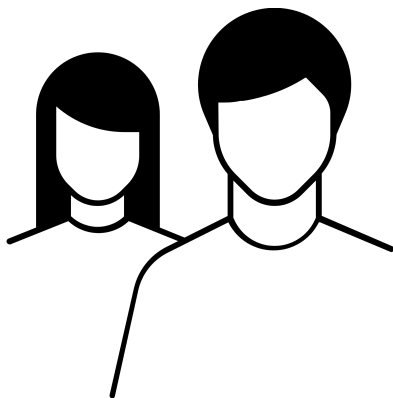
Home Connect in a Nutshell





Service Catalog v0:

People



- Scales terribly
- Memory persistence issues
- Bad version control
- Offline during holidays


Service Catalog v1:

Wiki

Pages / Home Connect: App & Services / Home Connect

HC Service Catalog

Created by Rüd. Trenzhaus (EQM42915), last modified by Chr. Peter (Sauer) on Oct 24, 2021



WOLFE RELEASE TRAIN

The **HC Service Catalog** is intended to give an overview about **all our HC-services** to people who are not daily working with a certain service, but need...

- general info (like ownership or deployment/filled status) or
- constructive links to certain details (like lock, docs, repository, handbook, interface)

Content

- Enduser Functions
- Content Editing
- HC Infrastructure
- HC Internal Services
- All Services

Enduser Functions

The following services provide functionality to the **enduser**:

Service	Ownership	Deployed Regions	Details
Account Parade Service (APCS)	IBM/AC/IBS	IBM/AC/IBS	
Android App	IBM	IBM/AC/IBS/UK/US	
Annotation Service	Team Razzak	Europe/UK/US	
Assisted Cooking Recipe Appliance Converter (BRAC)	Cookies	Europe/UK/US	
Assisted Cooking Widget	Cookies	Europe/UK/US	
Connectivity Based Sales	IBM/AC/IBS	Europe/UK/US	

- Not well integrated into workflows
- Distant from code
- In the end never up to date

Service Catalog v2:

Self-Built App

A screenshot of a web application interface for 'Services'. It features a blue header with a menu icon and the word 'Services'. Below the header, there's a 'Filter services:' section with a dropdown menu set to 'All Teams' and a search input field. A section titled 'Found 161 services:' displays a grid of service cards. Each card shows a service name (e.g., 'ac-mwa', 'ac-fbc', 'annotation-service-mongo-client', 'app-metrics-service', 'articles', 'articles-builder', 'assisted-cooking-widget-server', 'backstage', 'bean-equalizer-mongo-shell', 'bean-equalizer-server') and a 'MORE' link. The cards are arranged in two columns.

- High implementation and maintenance effort
- Unclear requirements
- Low buy-in from stakeholders & other teams

Backstage



Beyond the Microservice Honeymoon

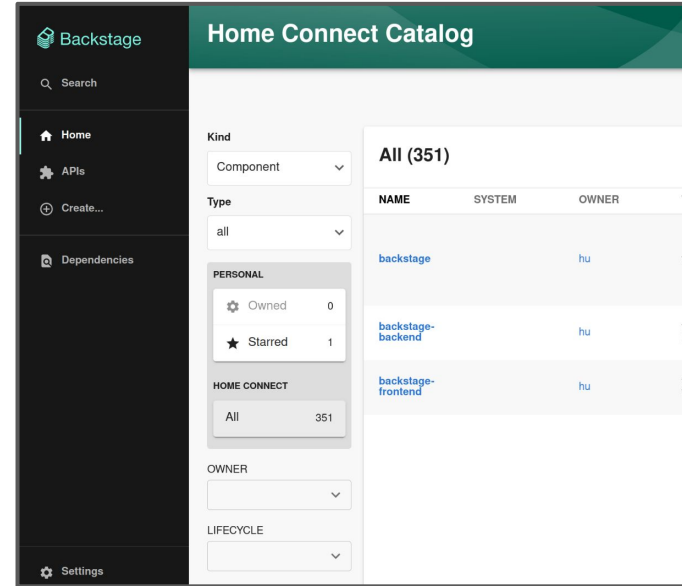


backstage.io

- Backstage addresses by
 - collecting service metadata & tools in one place
 - DRYing common microservice tasks

“Backstage - an open platform for building developer portals”

backstage.io



Goals



Reveal high-level architecture & dependencies



Give details about every service like ownership, docs, links, ...



Drive technology standardization



Speed up on-boarding

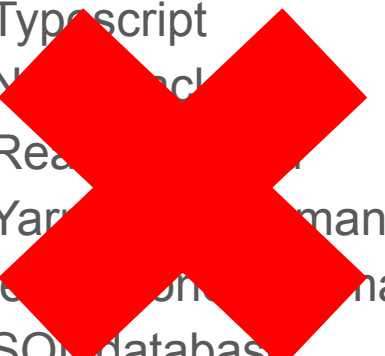


Reduce context switches for developers

Main Features

- Software Catalog
- API view including Swagger UI
- Search
- TechDocs
- Templates
- Highly extensible via front- & backend plugins

Tech Stack

- Typescript
 - Next.js
 - React
 - Yarn package manager
 - Lerna monorepo manager
 - SQL database
- 

⇒ What can Backstage tell us about Backstage?

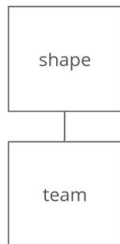
Introducing Backstage Step by Step

- Setup & deploy
- Create internal how-to, get data into Backstage
 - define data model, standards, & conventions
 - ongoing process to maintain data, push adoption
- Backstage declared official Home Connect service catalog
- Templates to quickly setup new services
- Custom plugin PoC: Dependency Search Tool
- Recently: modeling deployed instances & environments



Our Backstage Data Model

GROUPS



COMPONENTS



RESOURCES



APIS



Data Import

- Teams identify their entities (services, APIs, resources...)
- Add catalog-info.yaml files to the respective repos with
 - links to docs, other tools
 - ownership
 - relations to other services
- Auto-discovered by Backstage

```
! catalog-info.yaml x
microservice-demo > ! catalog-info.yaml > {} spec > {} def
You, 2 days ago | 1 author (You) | Catalog Info Backstage (c
1  apiVersion: backstage.io/v1alpha1
2  kind: Component
3  metadata:
4    name: microservice-demo
5    description: "Template for micro-se
6    links:
7      - url: https://microservices.ci.hcf
8        title: CI/CD build-job
9        icon: dashboard
10     - url: https://microservices.ci.hcf
11       title: CI/CD deployment-job
12       icon: dashboard
13  spec:
14    type: service
15    lifecycle: experimental
16    owner: microservice-cop
17    dependsOn:
18      - component:node-common
19    providesApis:
20      - microservice-demo-api
```



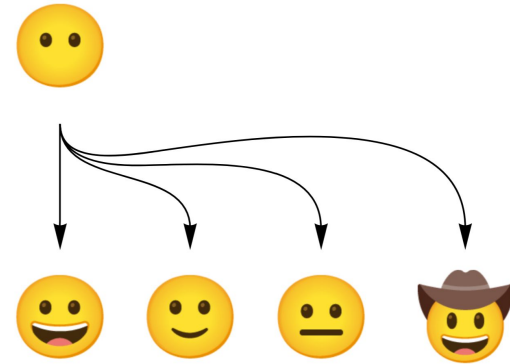
MICROSERVICE DEVELOPER

imgflip.com

JAKE-CLARK.TUMBLR

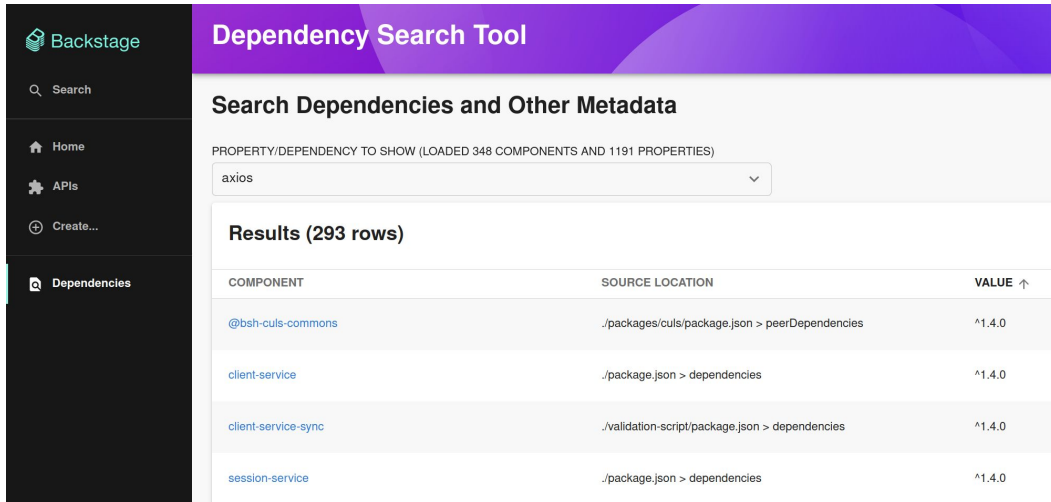
Templates

- Highly configurable backend actions with frontend form input
 - Default actions available for checking out repos etc
- Our templates
 - take in new service/repo name
 - copy repos, create default branches, permission settings
 - open PR in CI config repo
 - ...
- Demo



Custom Plugin: Dependency Search Tool

- PoC for plugin mechanism
- Data ingested via custom *processors*
- Filterable table view (react component) installed into main app
- In future teams shall develop their own plugins!



The screenshot displays the Backstage interface for the 'Dependency Search Tool'. The left sidebar contains navigation links for Home, APIs, Create..., and Dependencies. The main content area features a search bar with the text 'axios' and a dropdown arrow. Below the search bar, the text 'Results (293 rows)' is displayed. A table with three columns is shown: 'COMPONENT', 'SOURCE LOCATION', and 'VALUE ↑'. The table contains four rows of data:

COMPONENT	SOURCE LOCATION	VALUE ↑
@bsh-culs-commons	./packages/culs/package.json > peerDependencies	^1.4.0
client-service	./package.json > dependencies	^1.4.0
client-service-sync	./validation-script/package.json > dependencies	^1.4.0
session-service	./package.json > dependencies	^1.4.0

Needs to Be Customized for Your Organization

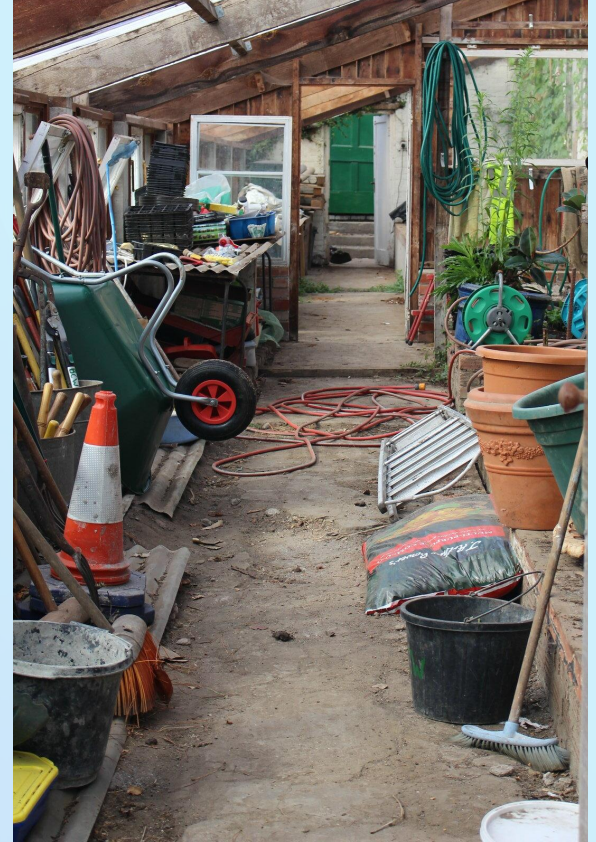
- By extending data model
- Scraping your info sources via *processors*
- Custom templates
- Custom front- and backend plugins
- ...

But: investment needed to build out customizations

Limitations

- Entity testing & error feedback difficult
- Many of the OSS plugins still in infancy
- Some features still in alpha

Take Aways



Why Did Backstage Work at Home Connect?

- Developers document right in their code bases
- Enough initial capability to get buy-in from stakeholders
- Extensible to cover needs of different user groups

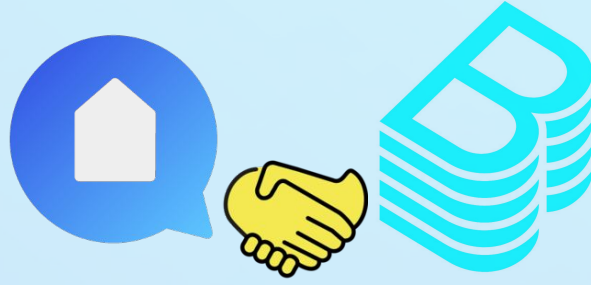
Is It For You?

Pro

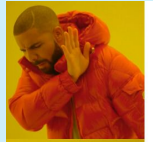
- > 10 of services/APIs
- > 3 teams
- Demo-repos → templates
- Cheap to try out
- Platform-team to take ownership

Con

- It's not free productivity
- Some features still in alpha



Thank you.



Inflation



Microservice
inflation