Towards customizing multi-tenant Cloud applications using non-intrusive microservices

Phu H. Nguyen, Hui Song, Franck Chauvel (SINTEF), Erik Levin (Visma)

Microservices Conference, Dortmund, February 19th, 2019
One of Europe’s largest independent research organisations

- 2000 Employees
- 75 Nationalities
- 4000 Customers

NOK 3.1 billion Revenues | NOK 450 MILL International sales

SINTEF
Close working relationship

Focus on:

- R&D Results, Patents and IP
- Products and Services
- System Solutions
- Business Concepts
- Spin-off Companies
Organization

SINTEF Building and Infrastructure
SINTEF Digital
SINTEF Materials and Chemistry
SINTEF Technology and Society
SINTEF Energy Research
SINTEF Fisheries and Aquaculture
SINTEF Petroleum Research
MARINTEK
SINTEF Digital

- Software and Service Innovation
- Software engineering, Safety and Security
- Mathematics and Cybernetics
- Smart Sensor Systems
- Microsystems and Nanotechnology
- Acoustics and Communication

- HCI – Human Computer Interaction
- SIS – Secure IoT Software
- SD – Smart Data
Secure IoT Software Research Group

Twitter @secure_iot_grp
This talk shows 1) the problem of customization; 2) our customization approaches; and 3) some lessons learned.

The problem of customization for multi-tenant SaaS

Customization with intrusive microservices or non-intrusive?

Some lessons learned
1) The problem of customization: Why Cirrus?

- More than 60% of businesses adapt standard systems software to their needs\(^1\)
- Major\(^2\) (potential) customers will never move to the cloud if the service cannot be customised to their specific needs

\(^2\) A major customer is typically a customer with more than hundred employees.
Enterprise software needs customisation

**Why**
- Every company is unique
- Need to integrate with other services
- Everything is changing fast

**How**
- UI re-design
- New business logic
- New data types

![Diagram showing standard executable code and custom code](image-url)
Cirrus project: SINTEF in collaboration with Super Office and Visma

![Cirrus Project Facts](image)

Funded by the Research Council of Norway + SuperOffice and Visma.

<table>
<thead>
<tr>
<th></th>
<th>no. of employees</th>
<th>revenue</th>
<th>no. of customers</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SuperOffice</strong></td>
<td>220</td>
<td>415 MNOK</td>
<td>10 000</td>
</tr>
<tr>
<td><strong>VISMA</strong></td>
<td>8 000</td>
<td>10 BNOK</td>
<td>500 000</td>
</tr>
<tr>
<td><strong>SINTEF</strong></td>
<td>2 082</td>
<td>2 936 MNOK</td>
<td>3 580</td>
</tr>
</tbody>
</table>

SINTEF is the largest independent research organisation in Scandinavia.
The Cirrus project aims to

Provide technology, tools and processes to allow customers to modify a multi-tenant cloud-based software-as-a-service with customised code without compromising the benefits of cloud computing.
Customization

Shared application, shared database

Standard executable code

UI

Client

Server

Custom code

Replaceable code

Insertable code
Tool-sets for customization

Programmer (developer)
Tech. Consultant
Admin user

- Visual Programming
- Pref's & settings
- Scripting
- APIs

Ease of use / Scalability

Cost of ownership

Deep Customization!
SalesForce's APEX language
SuperOffice's CRMScript
2) Our customization approaches based on Microservices

What does customization by microservices mean?

What does "intrusive microservice" mean?

Towards non-intrusive customization?
Customization by microservices

Code for customization is written and packaged as microservices that interact with the base system for customizing user interface (UI), business logic (BL), and database (DB).

Microservices for customisation purposes can be packaged and deployed (e.g., on containers) separately from the main product and each other (of different tenants).

Benefits:
• Isolation, which is important for multi-tenant context.
• Independent development, deployment, and operation.
• Technology stack freedom.
"Intrusive Microservices"

Standard product
- Microservice or monolithic
- hotplug

Custom code as a microservice
- Only the changed function
- Fine-grained replacement
- Not limited by APIs
- No direct access to the product database
- New view template to feed the standard HTML generator (Razor)
Intrusive custom code

Embedded interpreter with product context

Callback code / page query

Custom code

Main code

JavaScript

C#

How much partners can customise?

Full context (code-level customisation)

Limited context (API and services)

(small) code flow

(large) code flow

Full context

Limited context
Interactions Example

Video demo: https://www.youtube.com/watch?v=IluCeTHbcxc&t=6s
Customization using **intrusive microservices** has a main **drawback**: "intrusive" call-back code!

How to be non-intrusive? It depends a lot on what kind of architecture of the "base" system (main product)!

- Our industrial partner Visma is migrating their ERP system to microservices architecture, which would be much more "customization-friendly".
- We can orchestrate the non-intrusive customization using microservices via API Gateway pattern if the base system has microservices architecture.
Non-intrusive customization via API Gateway

IAM Service

Browser

WepApp UI (ASP.Net Core MVC)

MobileApp

API Gateways

Naming Service

Tenant X’s Microservices

Tenent Y’s Microservices

Event Bus

Main Product Application (API + DB)

NetFLIX

EUREKA

REST

REST

RabbitMQ

docker
eShopOnContainers reference application
(Development environment architecture)

Client apps
- eShop mobile app
  - Xamarin.Forms
  - C#
  - xPlat. OS:
    - iOS
    - Android
    - Windows
- eShop traditional Web app
  - HTML
- eShop SPA Web app
  - TypeScript/Angular

Docker Host

API Gateways / BFF
- Mobile Shopping
- Mobile Marketing
- Web Shopping
- Web Marketing

Identity microservice (STS+users)
  - SQL Server database

Catalog microservice
  - SQL Server database

Ordering microservice
  - Ordering API
  - Ordering BackgroundTasks

Basket microservice
  - Redis cache

Marketing microservice
  - MongoDB/CosmosDB
  - SQL Server

Locations microservice
  - MongoDB/CosmosDB

Event Bus

RabbitMQ

Azure Service Bus

https://github.com/dotnet-architecture/eShopOnContainers
Interactions in non-intrusive customization
3) Some lessons learned.
Summary and lessons learned

Intrusive customization for multi-tenant SaaS using microservices is feasible, even for monolith!

Microservices architecture would be more customization-friendly for multi-tenant SaaS! Non-intrusive customization using microservices is also feasible then.

Microservices-based approach is only one of the main approaches of the Cirrus project. There are other approaches, e.g., based on Scripting language.

There is no one silver bullet! We should choose what's best for a specific system, and business model.

Email: phu.nguyen [at] sintef.no
Customization Tools
Right Tool for the Right Job

Parallell stacks
- Internal APIs
  - Intrusive Microservices
  - Customized standard code
- External API's
  - REST, SOAP, webHooks
  - Connectors, DB Mirroring

InProduct
- Scripting
  - Custom Screens, Extra Tables
- Workflow Automation
  - Macros, Visual Programming

3rd party Platforms
- i.e Zapier, IFTTT
- Integration Platform
  - Action & Triggers

Administrator
- Configuration
  - Admin client

Consultant
- InProduct
  - Custom Screens, Extra Tables

Programmer

Cost of ownership

€

€€€€
The Custom Code Challenges with multi-tenancy

Isolation of custom code

Isolation of customers

“Private cloud”

Assimilation

Scripting engine

Add-in

Economy of Scale
Different procedures for:
- Standard Apps
- Custom Apps

Support for:
- New customers
- Existing customers
- Migrations

Application Services:
- Authentication
- Licensing
- Billing
- Education
- AI Services
- Logging
- Monitoring
- Testing
- etc.

For SuperOffice, this is part of the CRM Online Platform.
Main References & Acknowledgements

• The Cirrus project [https://www.sintef.no/en/digital/software-and-service-innovation/secure-iot-software/cirrus/](https://www.sintef.no/en/digital/software-and-service-innovation/secure-iot-software/cirrus/) This project has received funding from the Research Council of Norway, in collaboration with SuperOffice, and Visma.


This work is part of the Cirrus project funded by the Research Council of Norway, SuperOffice and Visma.
Technology for a better society