Microservices and GDPR at Sundhed.dk

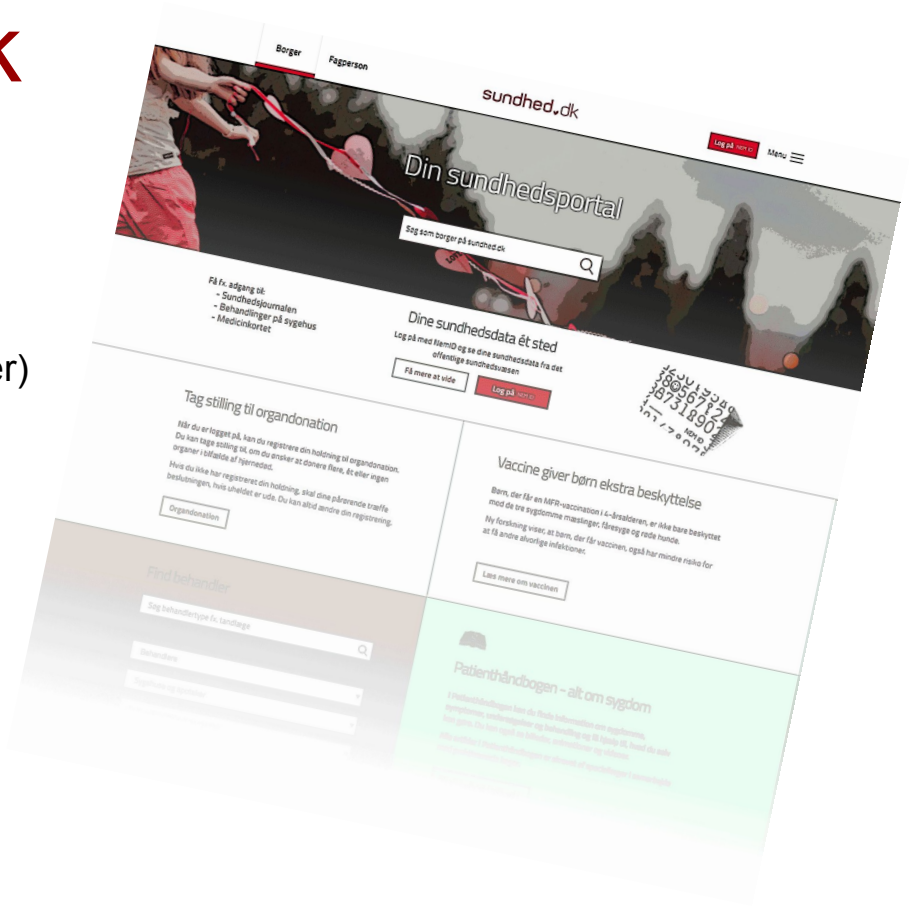
Tobias Uldall-Espersen
Thomas Holme

Background information:

Sundhed.dk and e-health in Denmark

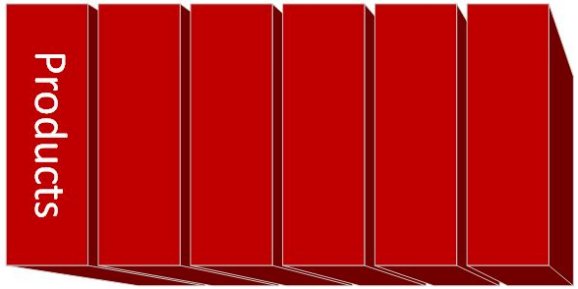
www.sundhed.dk

- Citizens, Healthcare professionals & Researchers
- CPR Numbers (Central Person Register)
- API’s & Gateways
- On-demand data access
  - vs data replication
- System centric vs Data Subject centric
- Microservices to scale software production and decouple products

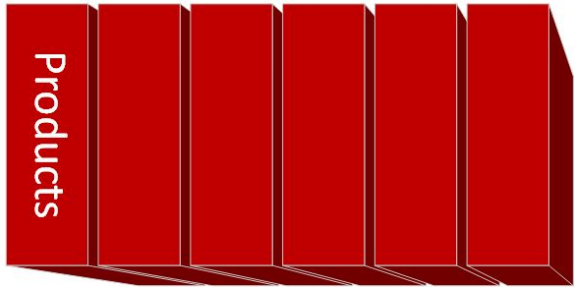


Introduction to ‘The Journey’

BUSINESS & LEGAL VIEW

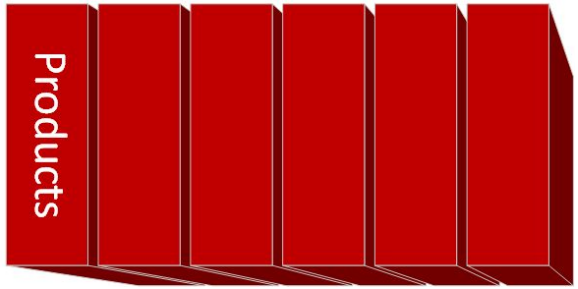


BUSINESS & LEGAL VIEW



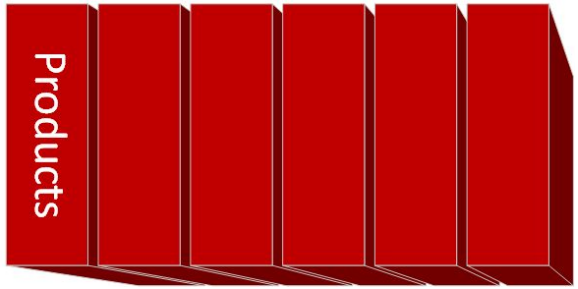
TECHNICAL VIEW

BUSINESS & LEGAL VIEW



TECHNICAL VIEW

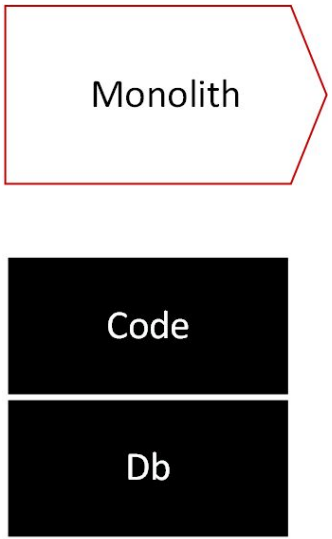
BUSINESS & LEGAL VIEW



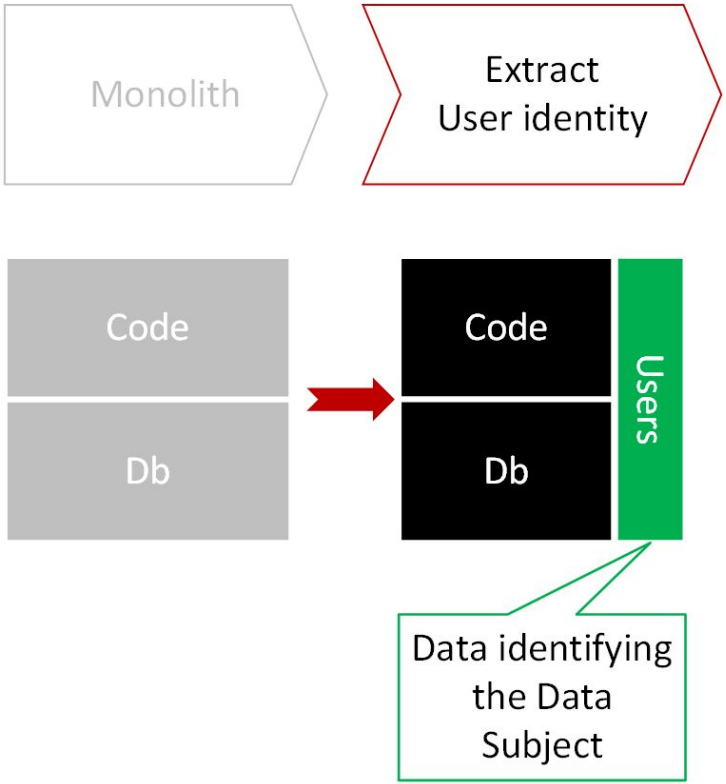
TECHNICAL VIEW

- CPR-Number, name, etc
- Data Subject identity with all data
- Trivial to Identify the Data Subject

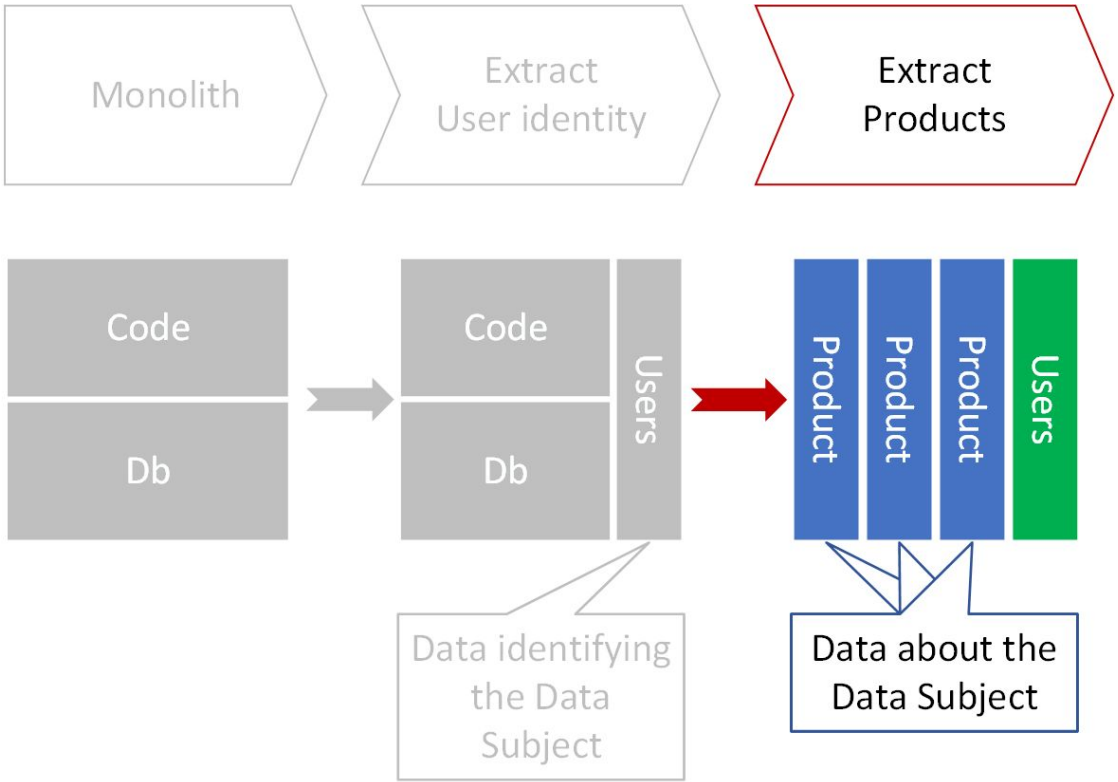
The transition - the starting point



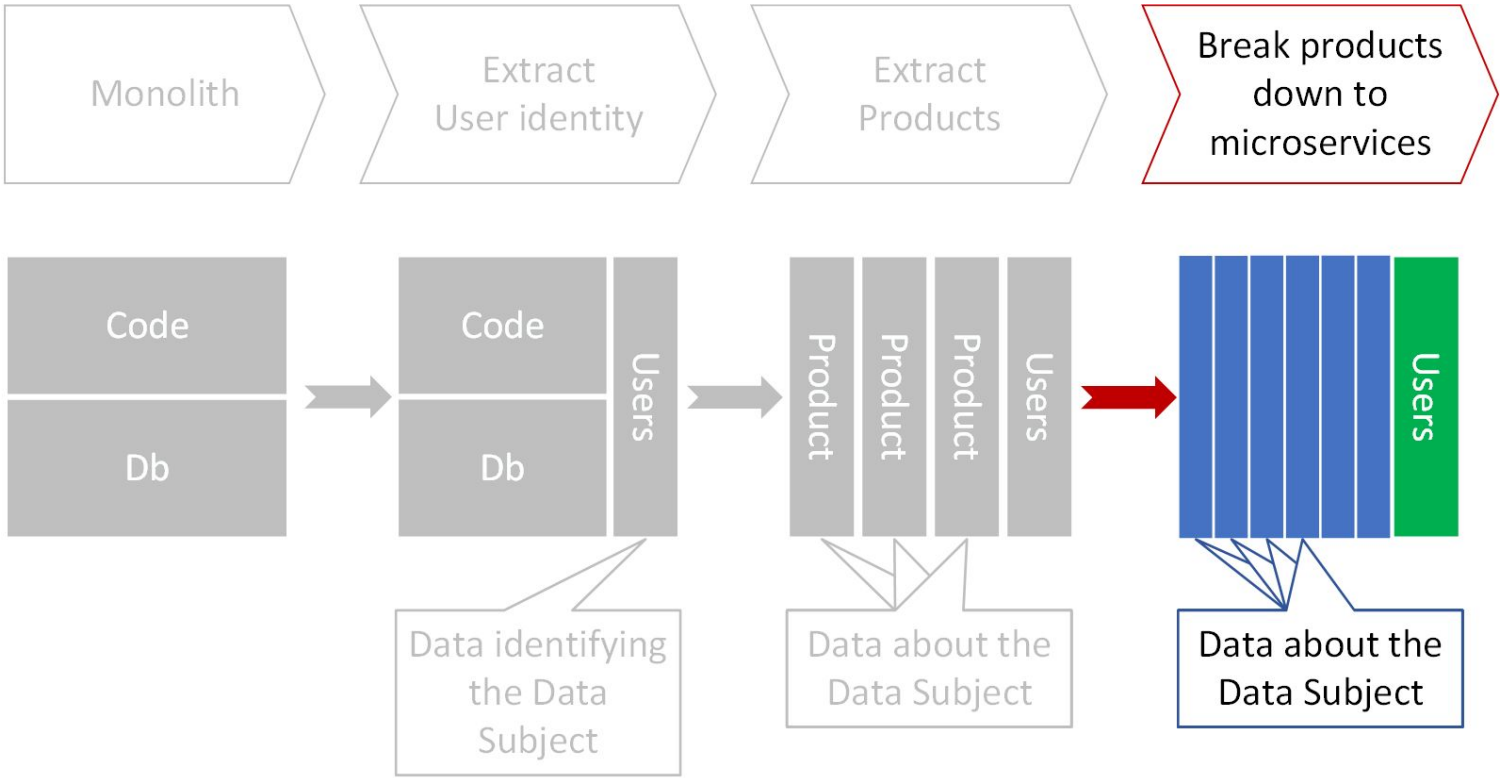
The transition - pseudomizing user identity in products



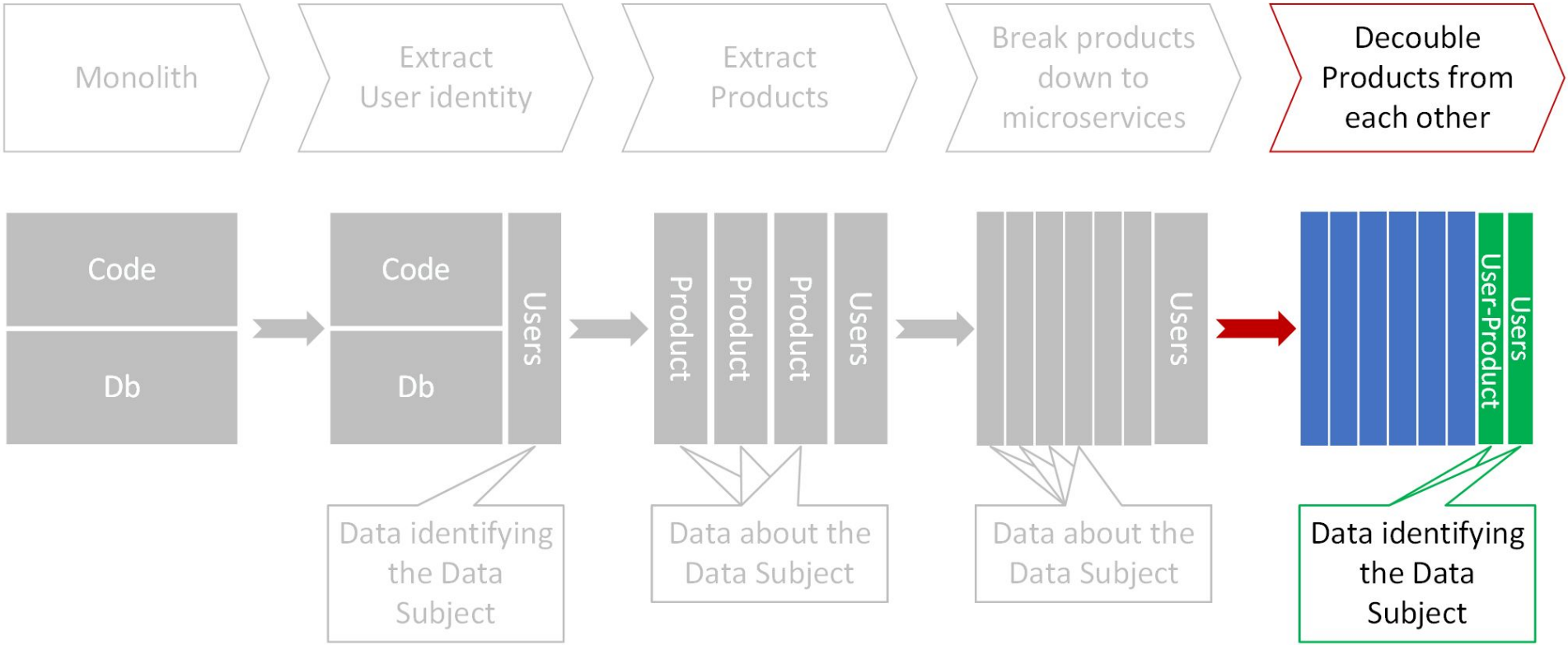
The transition - splitting the products from the monolith



The transition - microservices



The transition - decoupling across microservices



Background information: GDPR (a short introduction)

What is GDPR?

The General Data Protection Regulation (GDPR) (EU) 2016/679 is a regulation in **EU law on data protection and privacy** for all individuals within the European Union (EU) and the European Economic Area (EEA).

The General Data Protection Regulation consists of 99 articles and shall be applied from 25 May 2018

GDPR - Chapter 3: Rights of the Data Subject

- **Article 12 – Transparent information, communication and modalities for the exercise of the rights of the data subject**
- Article 13 – Information to be provided where personal data are collected from the data subject
- Article 14 – Information to be provided where personal data have not been obtained from the data subject
- **Article 15 – Right of access by the data subject**
- Article 16 – Right to rectification
- **Article 17 – Right to erasure (‘right to be forgotten’)**
- **Article 18 – Right to restriction of processing**
- Article 19 – Notification obligation regarding rectification or erasure of personal data or restriction of processing
- **Article 20 – Right to data portability**
- Article 21 – Right to object
- Article 22 – Automated individual decision-making, including profiling
- Article 23 – Restrictions

GDPR - Data protection by design and by default (Article 25)

1. Taking into account the state of the art, the cost of implementation and the nature, scope, context and purposes of processing as well as the risks of varying likelihood and severity for rights and freedoms of natural persons posed by the processing, **the controller shall, both at the time of the determination of the means for processing and at the time of the processing itself, implement appropriate technical and organisational measures**, such as **pseudonymisation**, which are designed to implement data-protection principles, such as **data minimisation**, in an effective manner and to integrate the necessary safeguards into the processing in order to meet the requirements of this Regulation and protect the rights of data subjects.

GDPR - Privacy by design (Article 25)

Privacy by Design: The 7 Foundational Principles

1. **Proactive not Reactive; Preventative not Remedial:** Privacy by Design anticipates and prevents privacy invasive events before they happen.
2. **Privacy as the Default Setting:** Privacy by Design deliver the maximum degree of privacy by ensuring that personal data are automatically protected in any given IT system or business practice
3. **Privacy Embedded into Design:** Privacy by Design is embedded into the design and architecture of IT systems and business practices. It is not bolted on as an add-on, after the fact.

(Source: Ann Cavoukian, Privacy by Design: The 7 Foundational Principles)

GDPR - Privacy by design (Article 25)

Privacy by Design: The 7 Foundational Principles

4. **Full Functionality – Positive-Sum, not Zero-Sum:** Privacy by Design seeks to accommodate all legitimate interests and objectives in a positive-sum “winwin” manner, not through a dated, zero-sum approach, where unnecessary trade-offs are made.
5. **End-to-End Security – Full Lifecycle Protection:** Privacy by Design, having been embedded into the system prior to the first element of information being collected, extends securely throughout the entire lifecycle of the data involved

(Source: Ann Cavoukian, Privacy by Design: The 7 Foundational Principles)

GDPR - Privacy by design (Article 25)

Privacy by Design: The 7 Foundational Principles

6. **Visibility and Transparency – Keep it Open:** Privacy by Design seeks to assure all stakeholders that whatever the business practice or technology involved, it is in fact, operating according to the stated promises and objectives, subject to independent verification.
7. **Respect for User Privacy – Keep it User-Centric:** Above all, Privacy by Design requires architects and operators to keep the interests of the individual uppermost by offering such measures as strong privacy defaults, appropriate notice, and empowering user-friendly options.

(Source: Ann Cavoukian, Privacy by Design: The 7 Foundational Principles)

GDPR - Privacy Design Strategies

Data oriented strategies

a. **MINIMISE** - The amount of personal data that is processed should be restricted to the minimal amount possible.

b. **HIDE** - Any personal data, and their interrelationships, should be hidden from plain view.

c. **SEPARATE** - Personal data should be processed in a distributed fashion, in separate compartments whenever possible.

d. **AGGREGATE** - Personal data should be processed at the highest level of aggregation and with the least possible detail in which it is (still) useful.

(Source: Jaap-Henk Hoepman)

GDPR - Privacy Design Strategies

Process oriented strategies

a. **INFORM** - Data subjects should be adequately informed whenever personal data is processed.

b. **CONTROL** - Data subjects should be provided agency over the processing of their personal data.

c. **ENFORCE** - A privacy policy compatible with legal requirements should be in place and should be enforced.

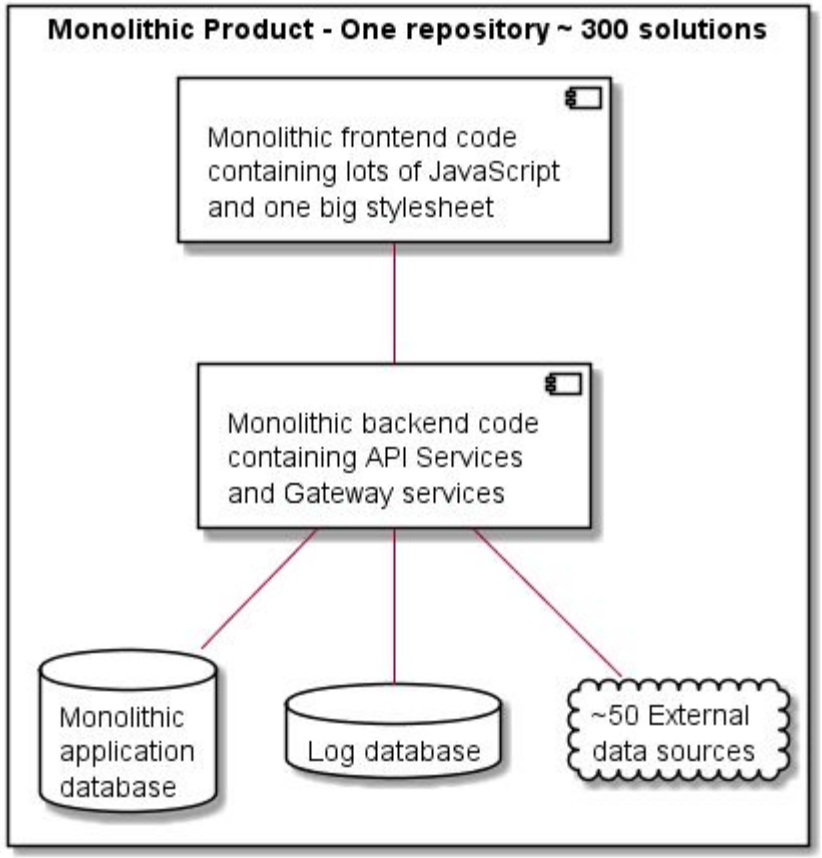
d. **DEMONSTRATE** - Be able to demonstrate compliance with the privacy policy and any applicable legal requirements

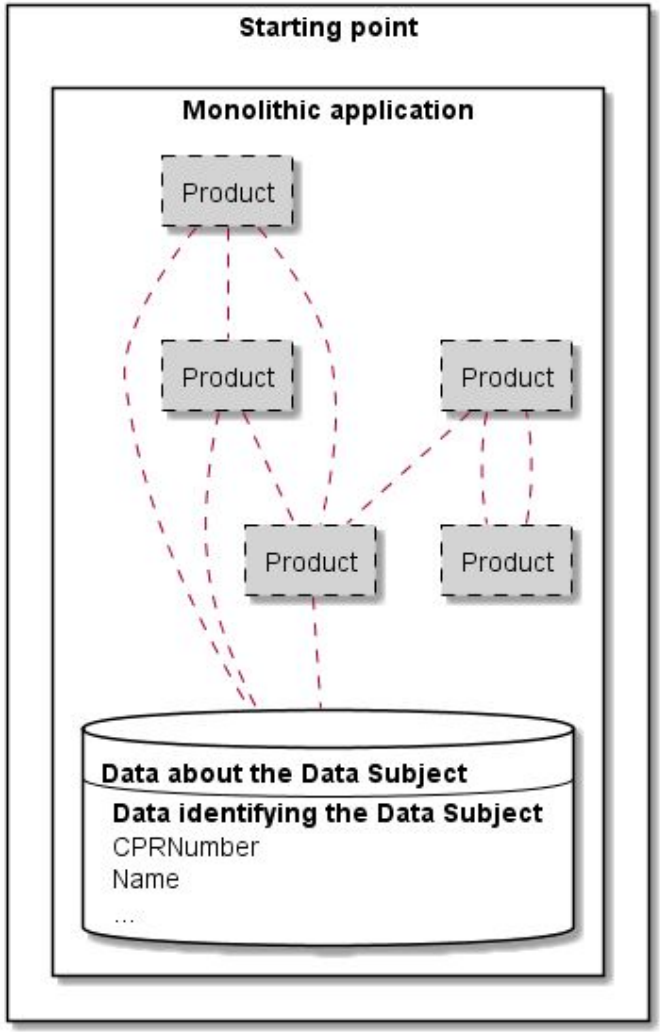
(Source: Jaap-Henk Hoepman)

GDPR and Microservices at Sundhed.dk

1. Pseudonymising data in the monolith
2. Separating data into data contexts with single ownership
3. Making it User-Centric

First step:
Pseudonymising data in the monolith

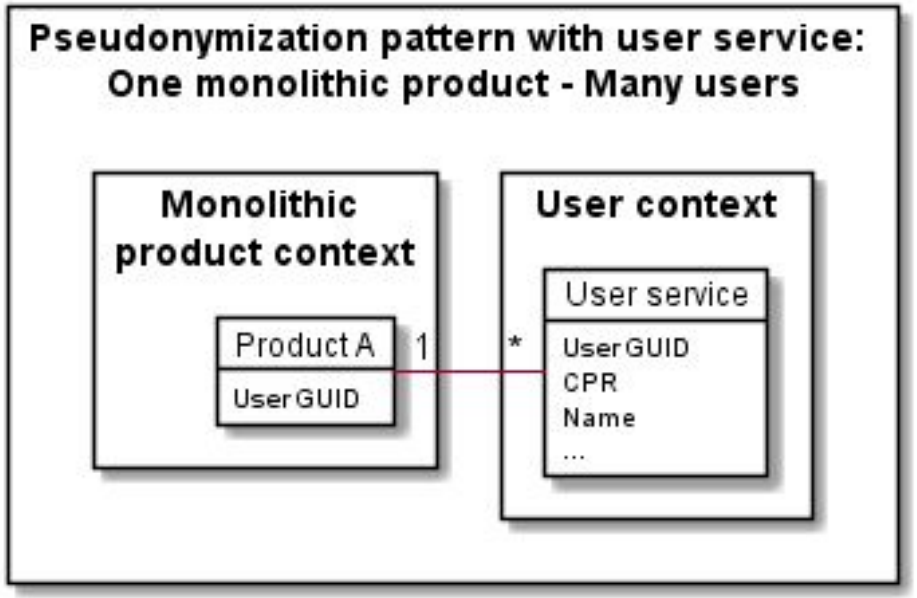

Monolithic Product - One repository ~ 300 solutions
Monolithic frontend code containing lots of JavaScript and one big stylesheet
Monolithic backend code containing API Services and Gateway services
Monolithic application database
Log database
~50 External data sources


Starting point
Monolithic application
Product
Product
Product
Product
Product
Data about the Data Subject
Data identifying the Data Subject
CPRNumber
Name
...

First microservice

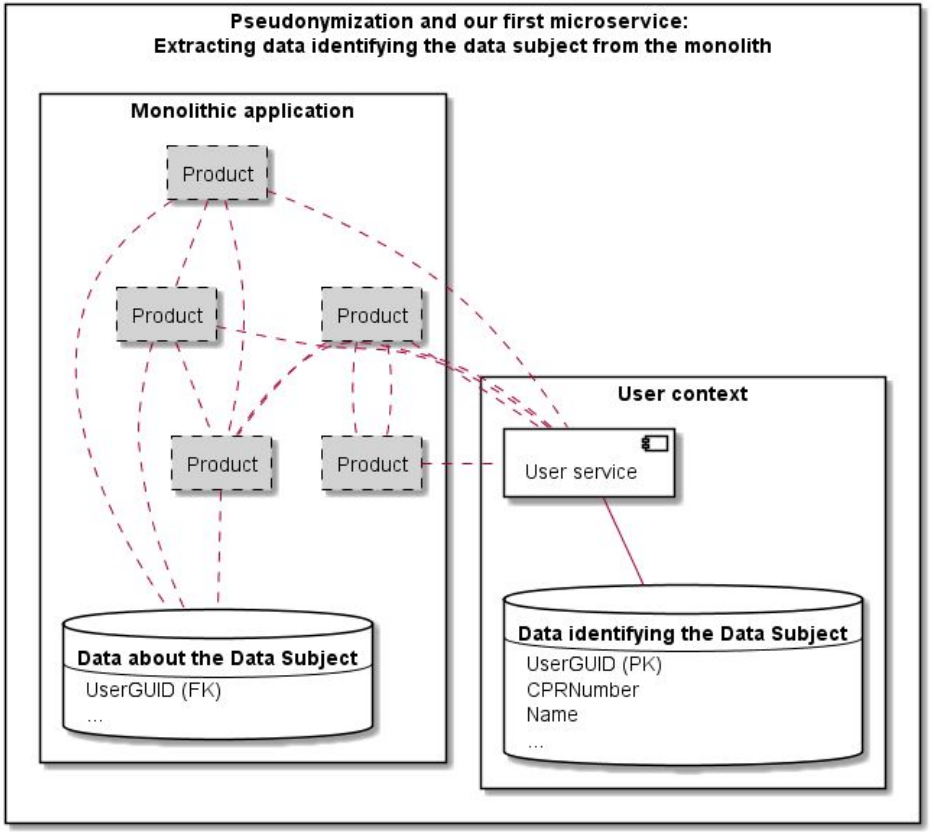
1:M relationship between product and user.

First real microservice.



Pseudonymizing data

Pseudonymizing data by splitting data identifying the data subject from data about the data subject.



Summing up: What we did

- Implemented initial microservice architecture
- Split data identifying the data subject from data about the data subject into two separate databases
- Generate an artificial (pseudonymization) key (UserGUID) binding the two set of data together
- Remove confidential data (CPR number) from products

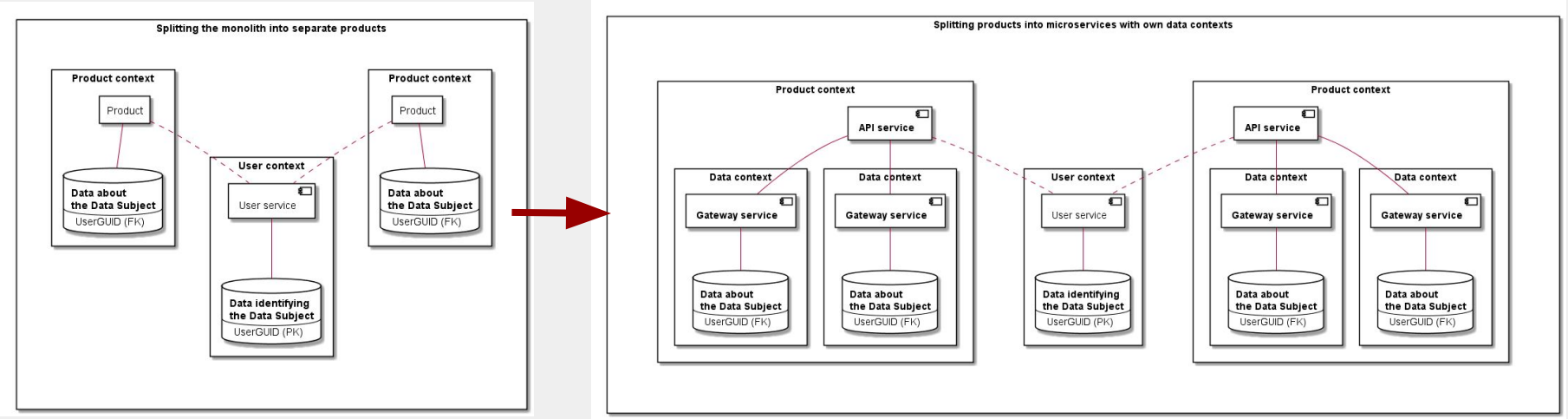
Summing up: What we gained

- Limiting processing of data identifying the data subject
- Portal global 'Right to be forgotten'
- Protection of confidential data (CPR)

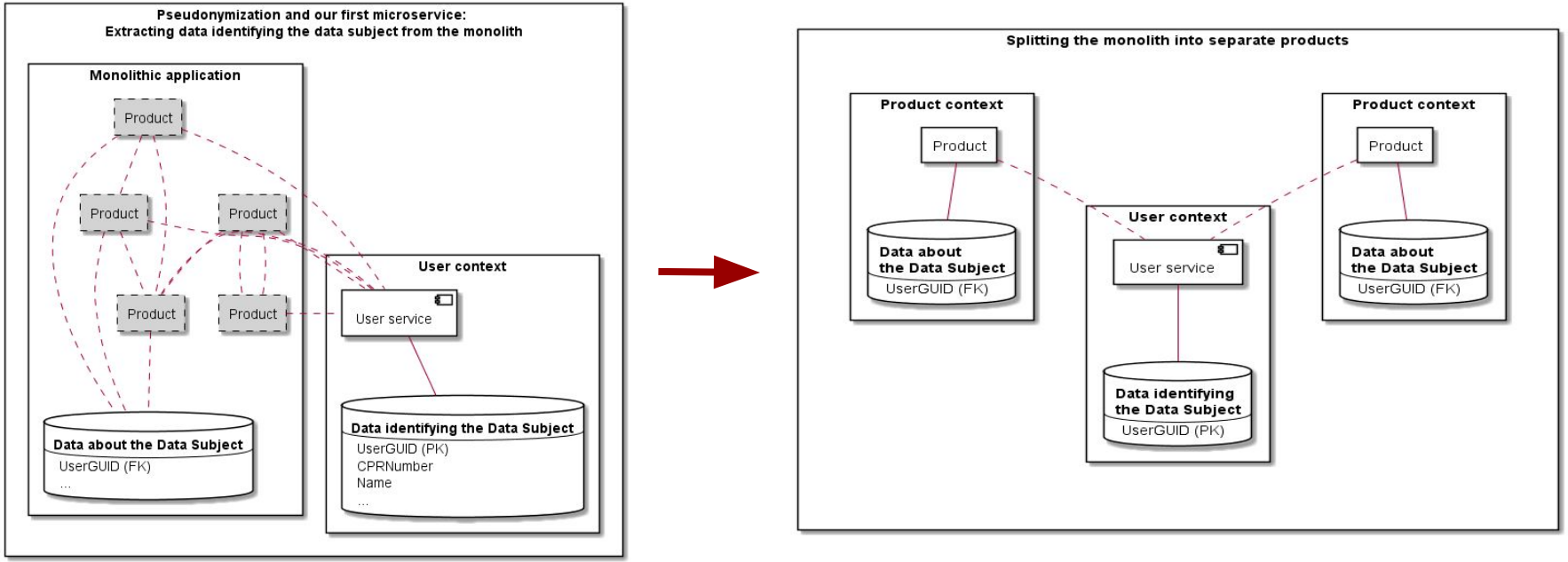
Summing up: Limits of current solution

- Possible to link product data and data subject data when having database access
- Possible to combine product data and create 'rich picture' of data subjects when having database access

Step 2: Separating data
First into product contexts
then into data contexts

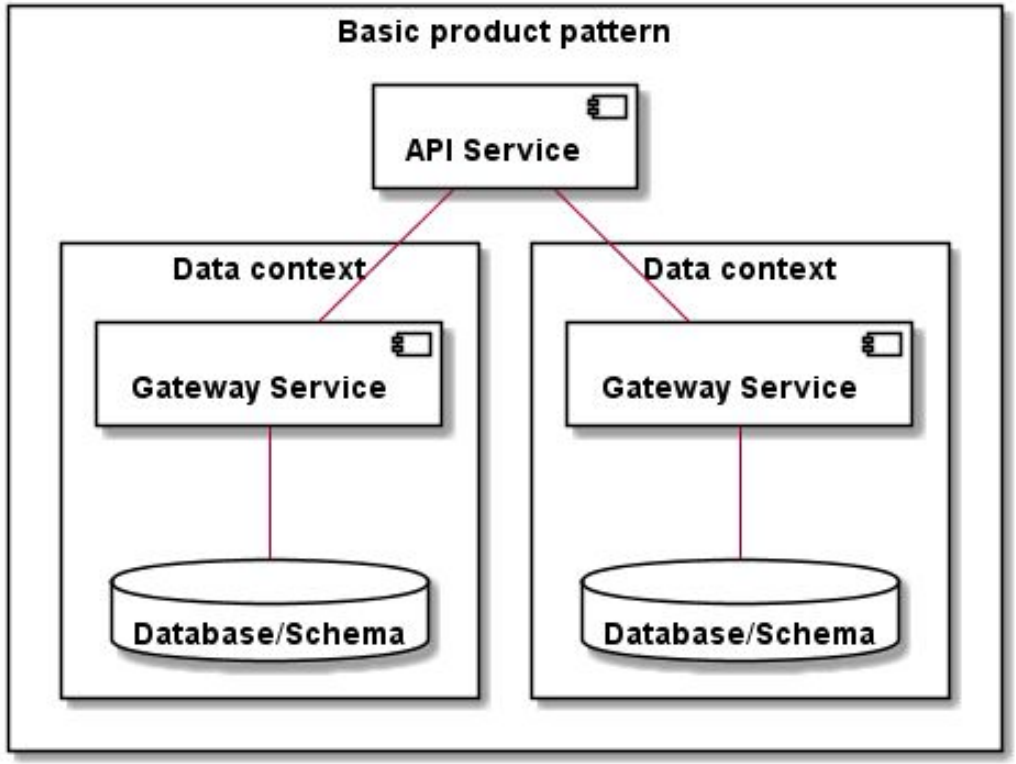


Extract each product with its data from the monolith



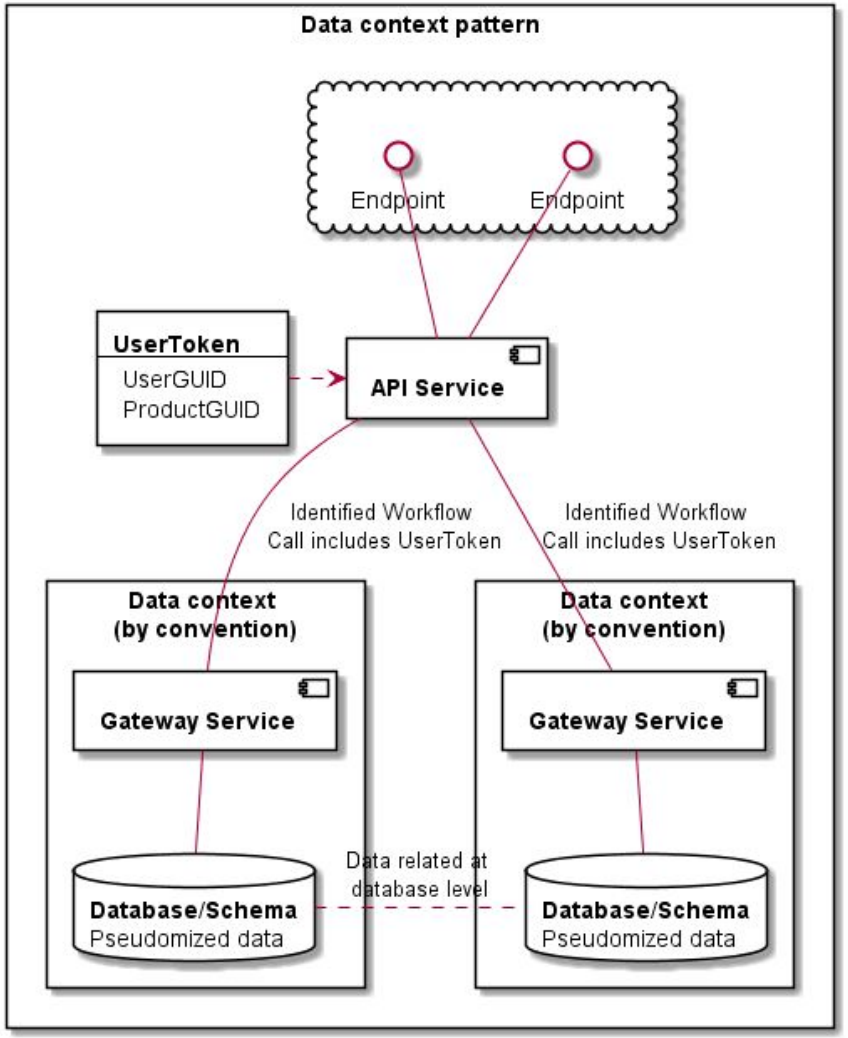
Each product is further divided:

- An API facade like service
- One or more underlying Gateway services.



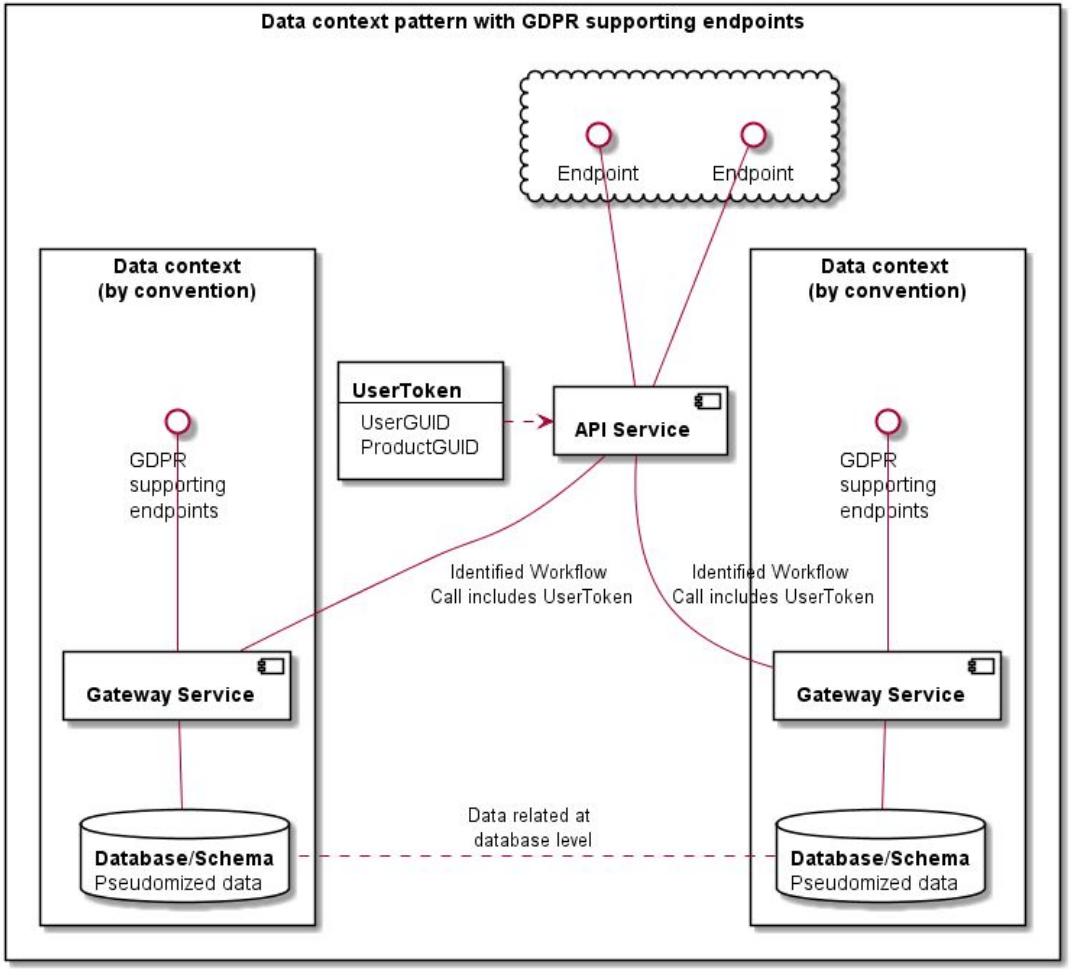
Ownership by convension

Gateways could acces data from one another, but by convention we agree no to.



GDPR endpoints

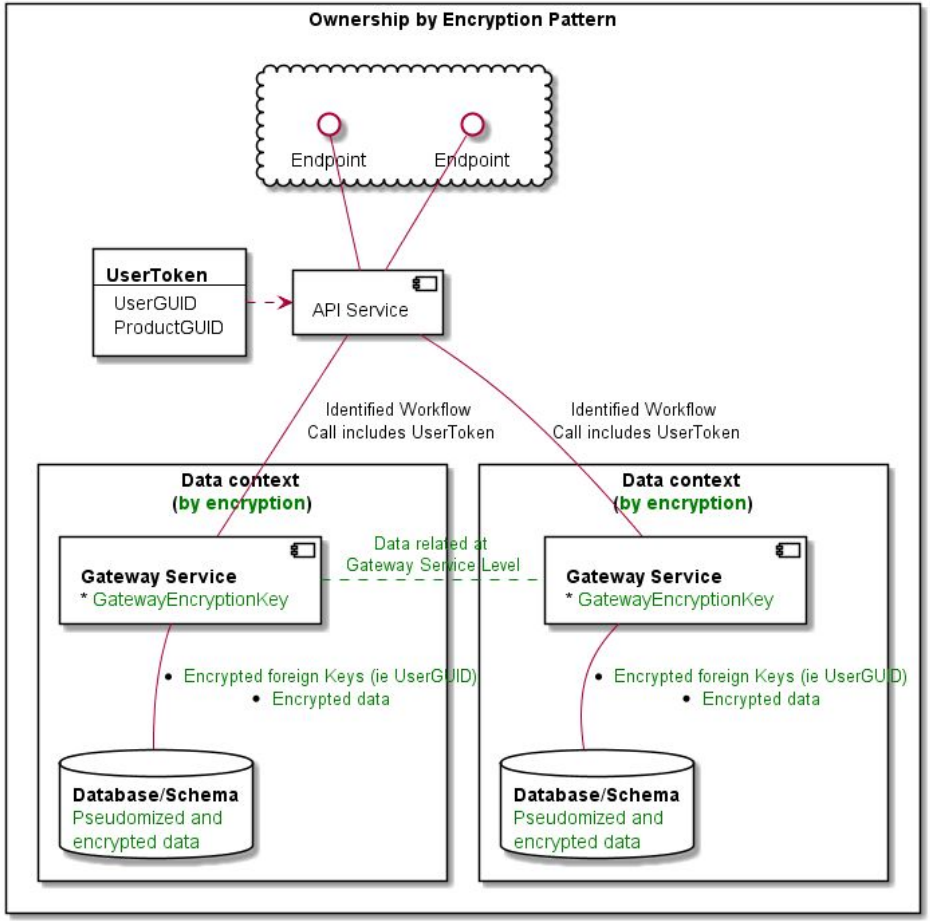
- **Right to rectification**
  ‘the right to get errors fixed’
  (Article 16)
- **Right to erasure**
  'the right to be forgotten'
  (Article 17)
- **Right to data portability**
  ‘the right to get a copy’’
  (Article 20)



Ownership by encryption

Data context is enforced by encryption rather than by convention.

The database content becomes nonsense to all but the owner.



Summing up: What we did

- Separated products from each other
- Separated contexts in each product from each other
- Added ownership of data to each context
- Added context specific encryption key to each context
- Encrypted pseudomization key (UseGUID) with context specific encryption key when used as foreign key
- Encrypted foreign keys and sensitive/confidential data with context specific encryption key

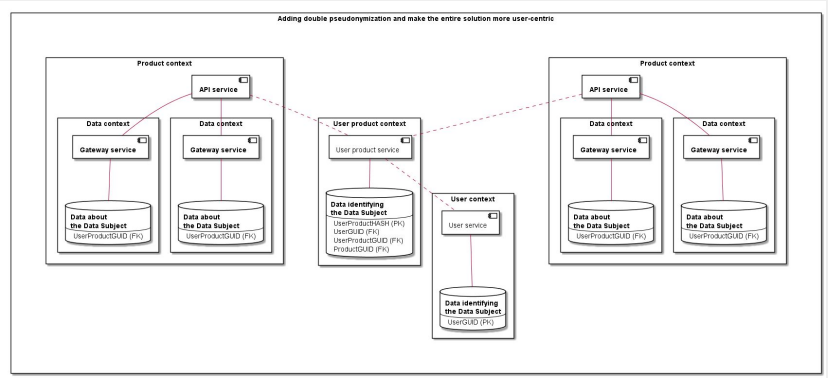
Summing up: What we gained

- Formalized data context with forced owner
- Simple contexts increases transparency and makes compliance control easier
- Avoid direct linking of data in the database
- Better support for logging
- Export of data build into context
- Better protection of data in entire data lifecycle

Summing up: Limits of current solution

- Not user centric solution
- Not possible to control individual products, e.g.in regards to:
  - Consent
  - Right to be forgotten
  - Limiting processing
  - Efficient access control supported by architecture

Step 3:
Making it User-Centric



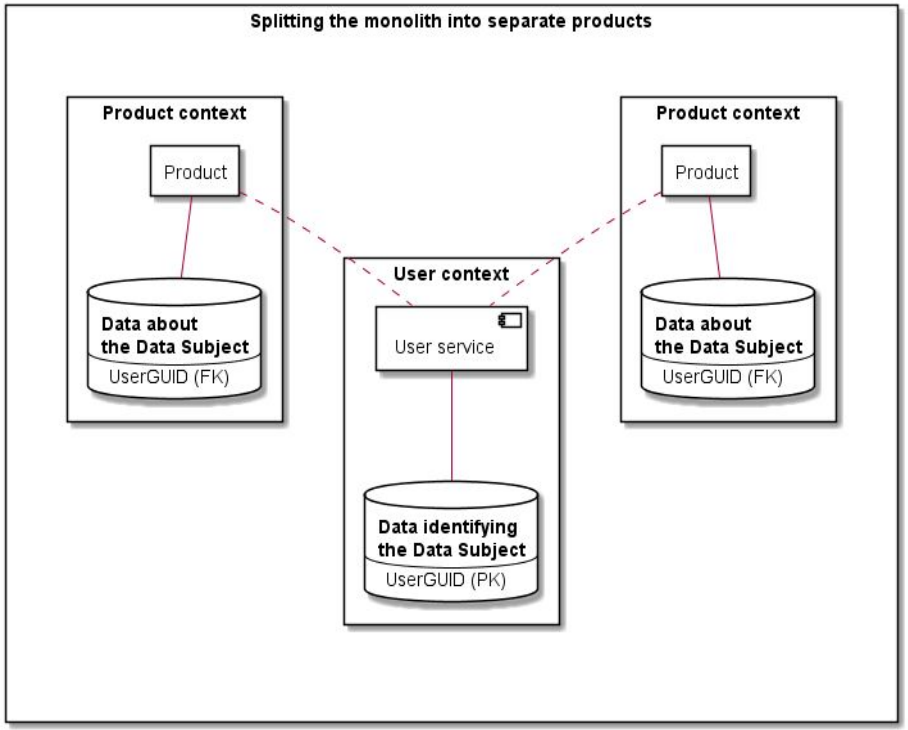
What does it mean, User-Centric?

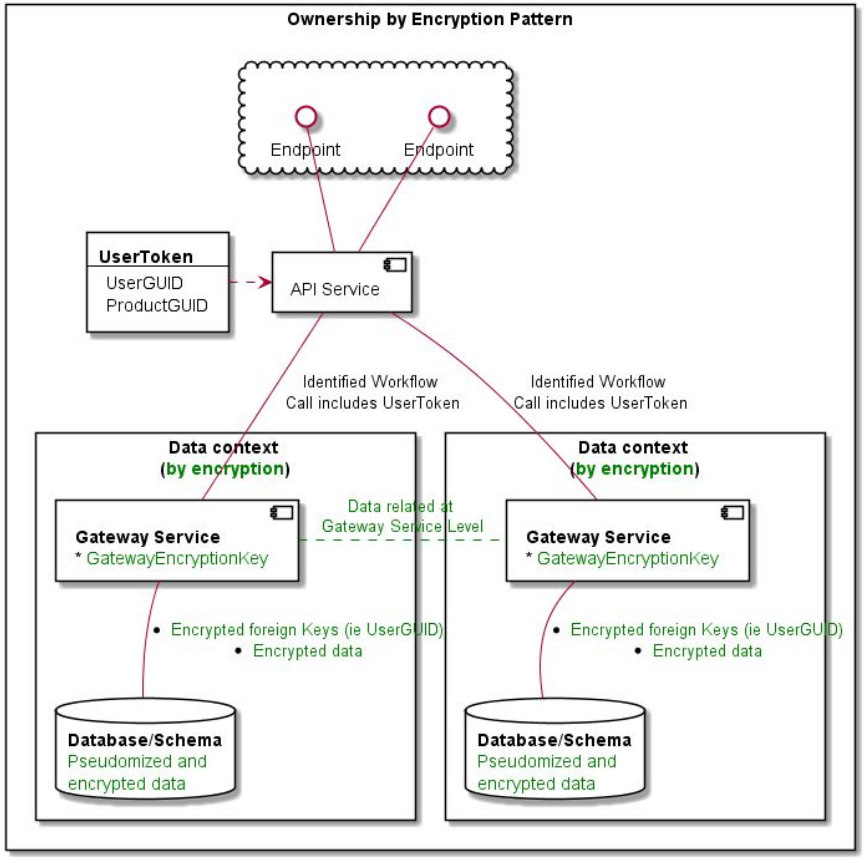
The root principle of privacy by design is based on **enabling service without data control transfer** from the citizen to the system.

Above all, Privacy by Design requires **architects** and operators to keep **the interests of the individual** uppermost by offering such measures as **strong privacy defaults**, appropriate notice, and **empowering user-friendly options**.

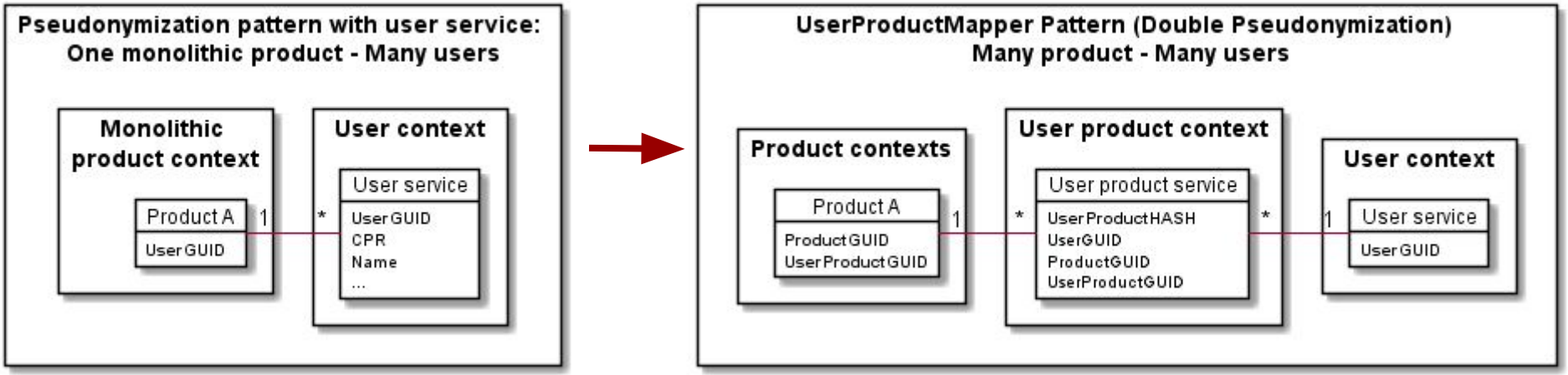
(Source: Ann Cavoukian, Privacy by Design: The 7 Foundational Principles)

Is this design user-centric?

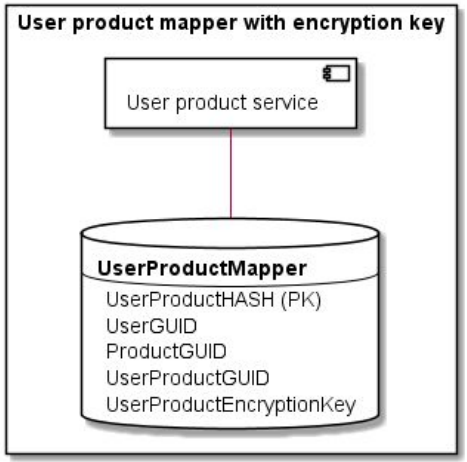


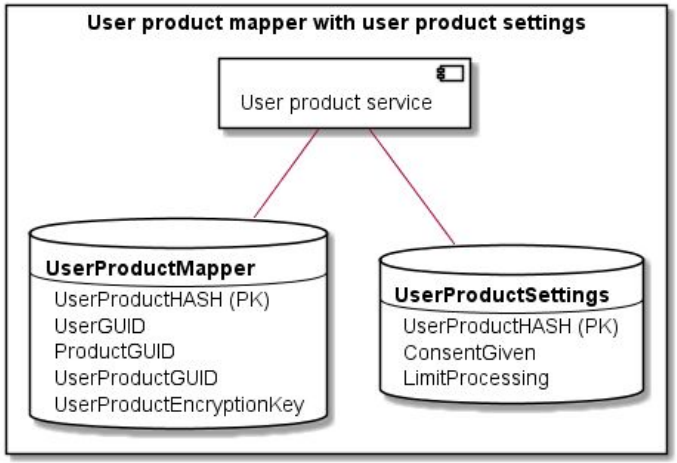


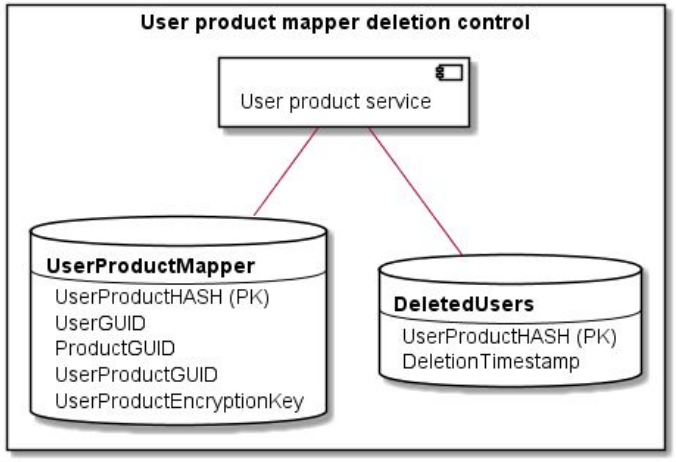
Adding multiple products to the design



Some different design considerations







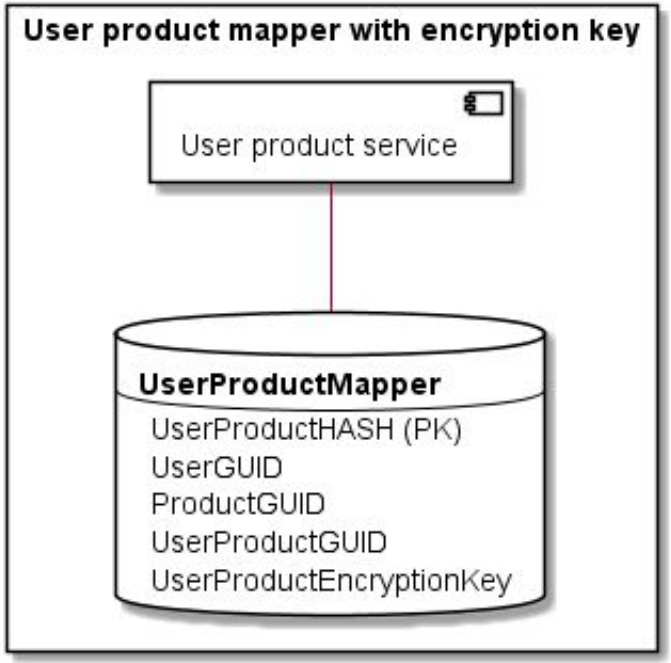
Adding a UserProductEncryptionKey

UserProductGUID used as double pseudonymized data subject identifier.

Providing better way to control the data subjects relationship to individual products.

UserProductEncryptionKey (UPEK) used for encryption of sensitive data.

UPEK becomes the key for letting the user control access to sensitive data.

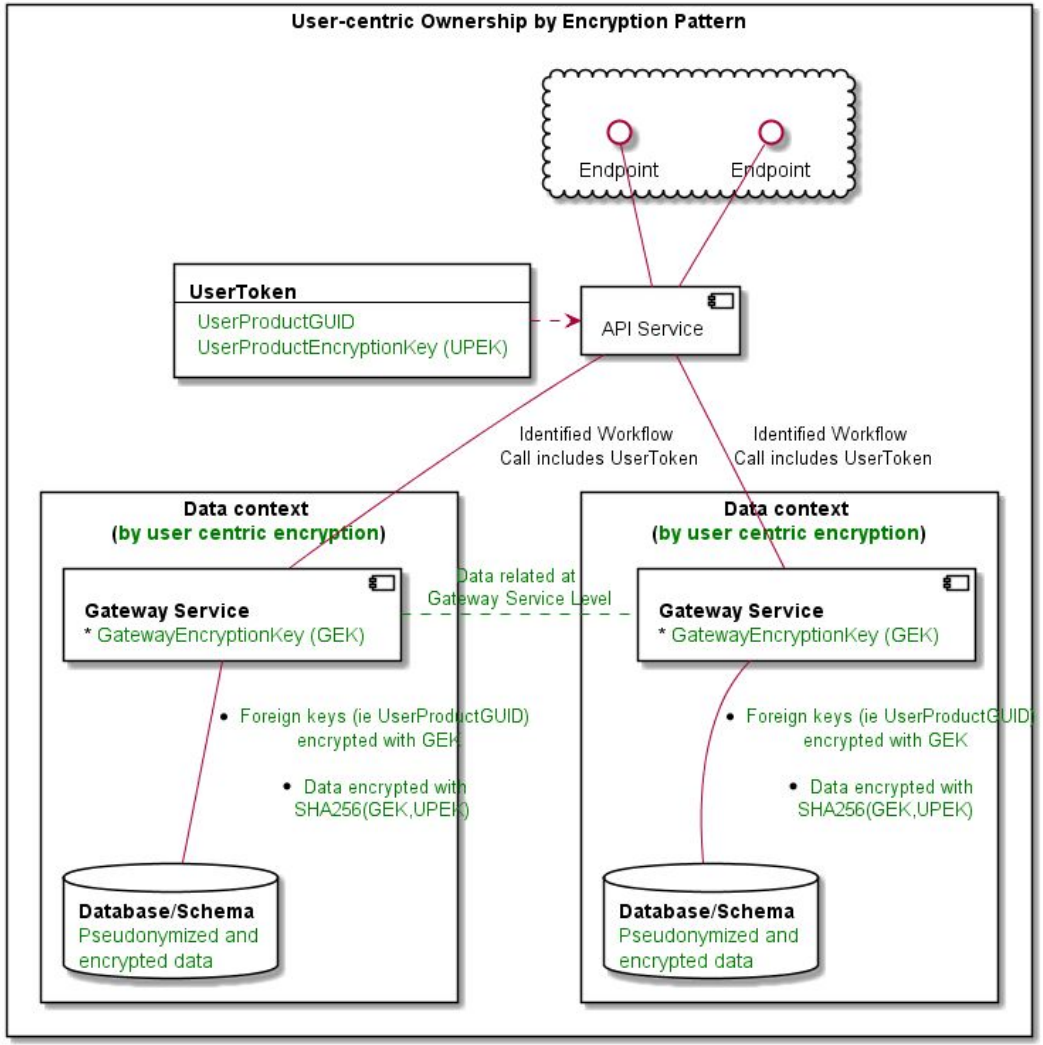


User-Centric Ownership by Encryption Pattern

UserProductGUID and UserProductEncryptionKey added to UserToken.

Sensitive data encrypted using hash of GatewayEncryptionKey and UserProductEncryptionKey.

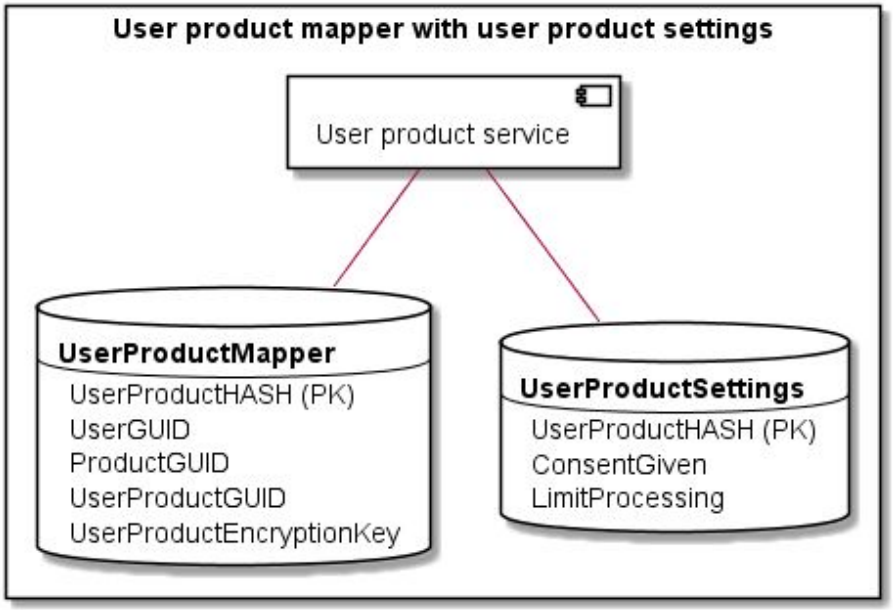
UserProductGUID used as foreign key.



Adding user product settings

Data subjects must give consent to each product. This increase transparency and makes system more user-centric.

Data subjects have a right to limit processing of data and thereby limiting other persons access to the data subjects data.



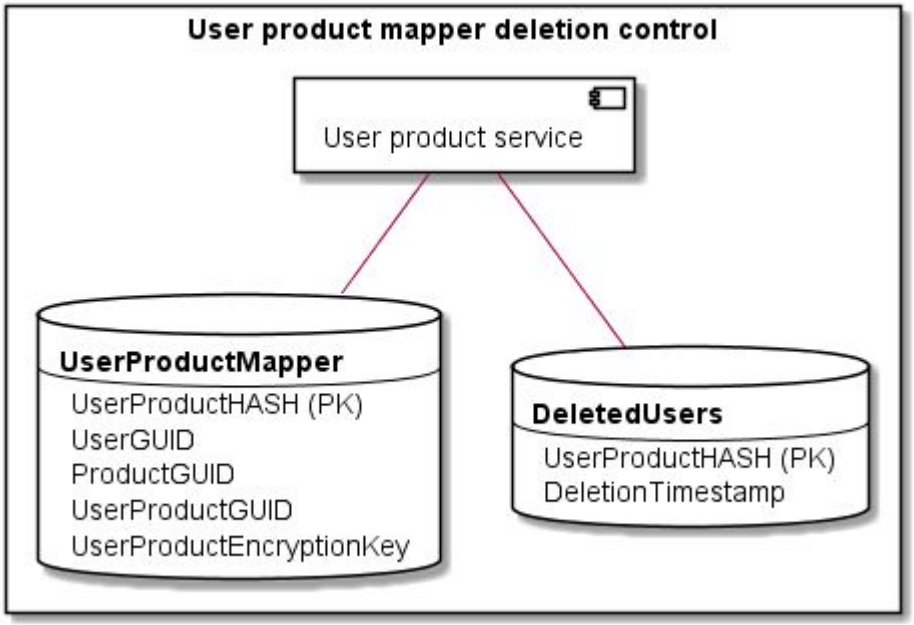
Adding deletion control

Deleting data across many microservices can be a challenge

Right to be forgotten - what about backups?

Delete or prevent restore…

Note that UserProductHASH is anonymous when UserProductMapping is deleted.



Summing up: What we did

- Added UserProductGUID and UserProductEncryptioKey to UserToken
- Created composite encryption key for sensitive data based og GatewayEncryptionKey and UserProductEncryptioKey
- Replaced UserGUID with UserProductGUID as foreign key and main user identifier in product database
- Added anonymous tracking of deleted users
- Added user specific product settings per product

Summing up: What we gained

- Better per user separation of data in rest, e.g. when storing and caching data encrypted with UserProductEncryptionKey
- Better separation when processing individual products (UserProductGUID)
- Centralized and standardized control of data access - no access to UserProductGUID and UserProductEncryptionKey means no access to data subject data. Privacy build into the Design.
- Better ways of deleting data subject data - deleting UserProductGUID makes product data unlinkable, deleting UserProductEncryptionKey makes sensitive data unreadable. (Positive Sum game)
- Better end-to-end security and privacy through entire life-cycle
- Generic, reusable and quite simple solution - not much additional complexity and cost

Summing up: Limits of current solution

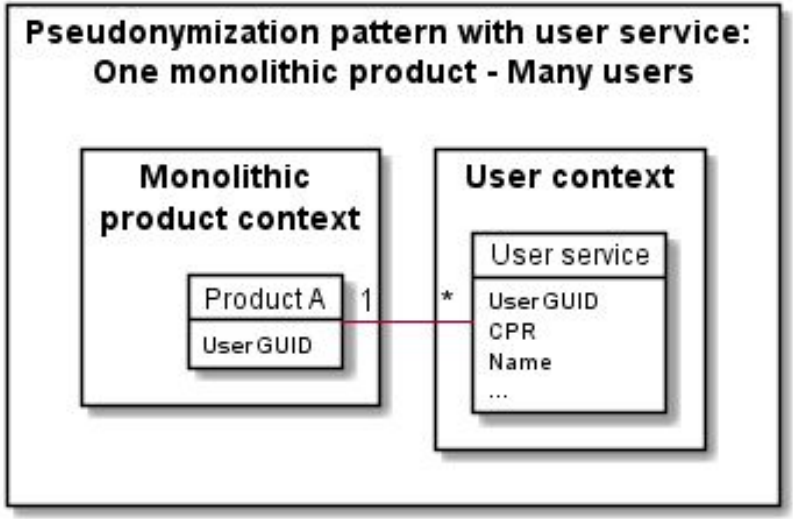
- Data access across users becomes more difficult
- Data access across product becomes more difficult

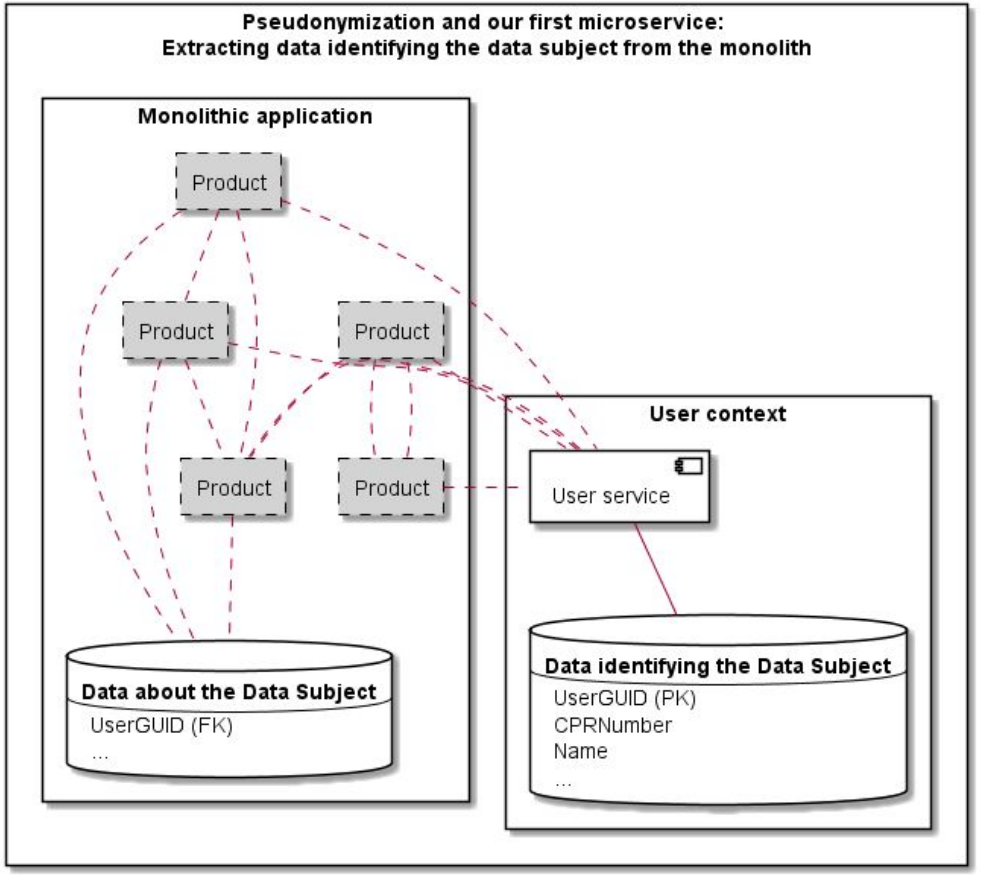
Lessons learned

- Three recommended patterns
- Final conclusions

Summarizing design consideration: Three recommended patterns

Data pseudonymization pattern





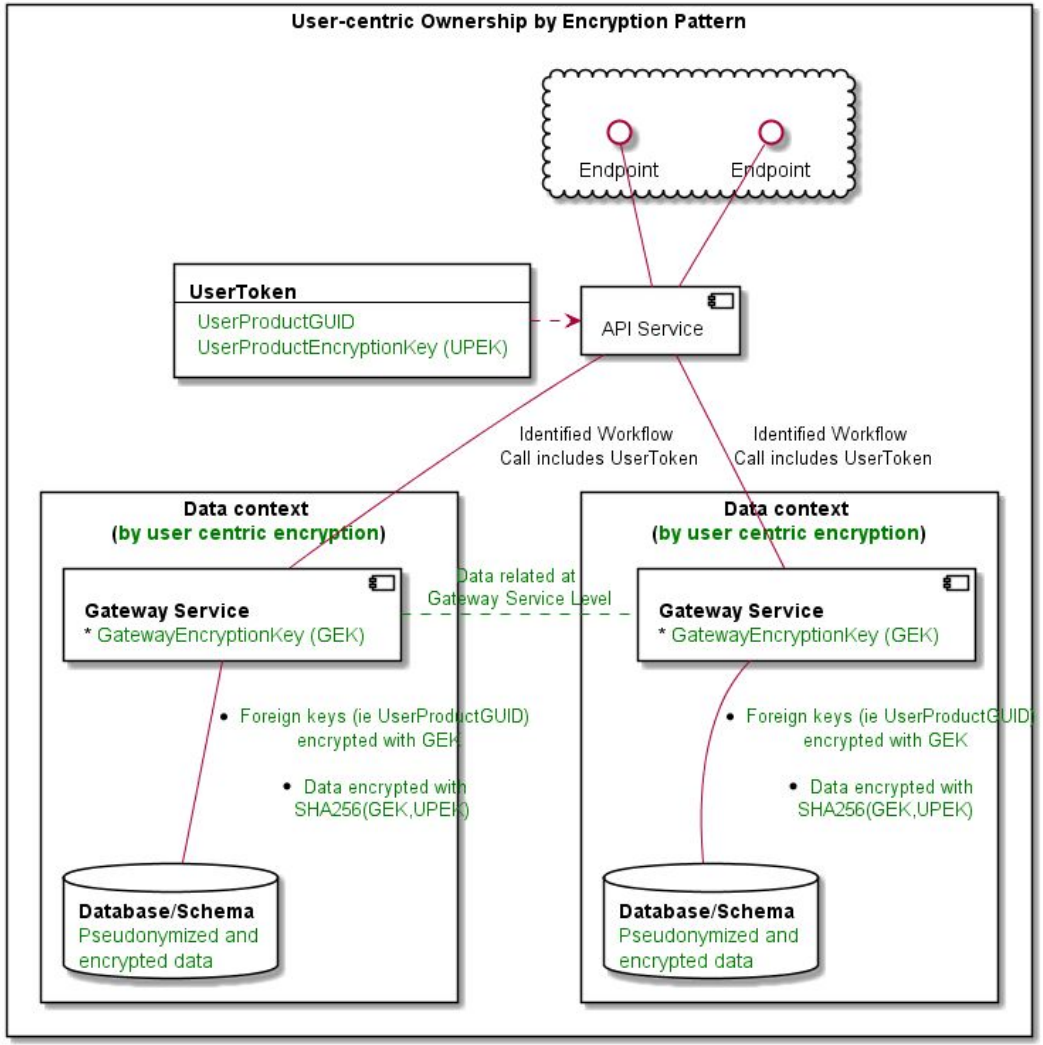
User-Centric Ownership by Encryption Pattern

Better separation of data

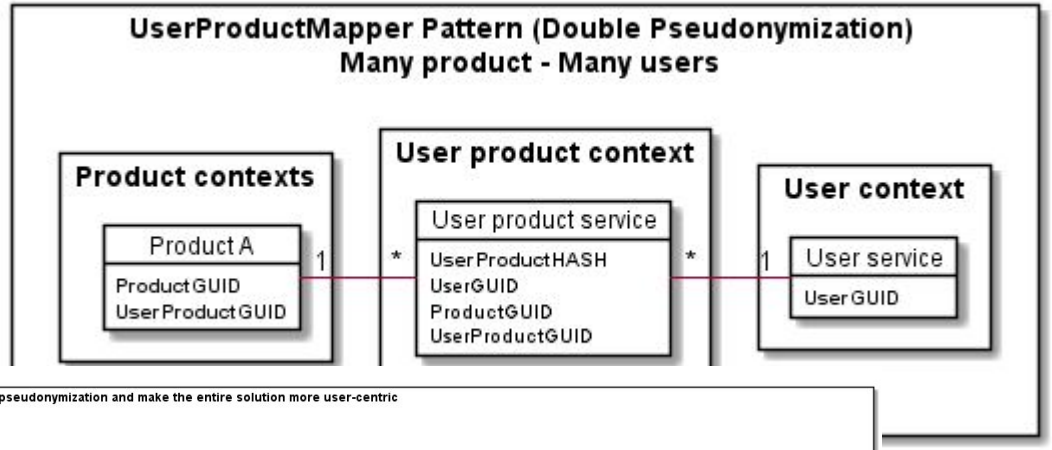
Stronger ownership of data.

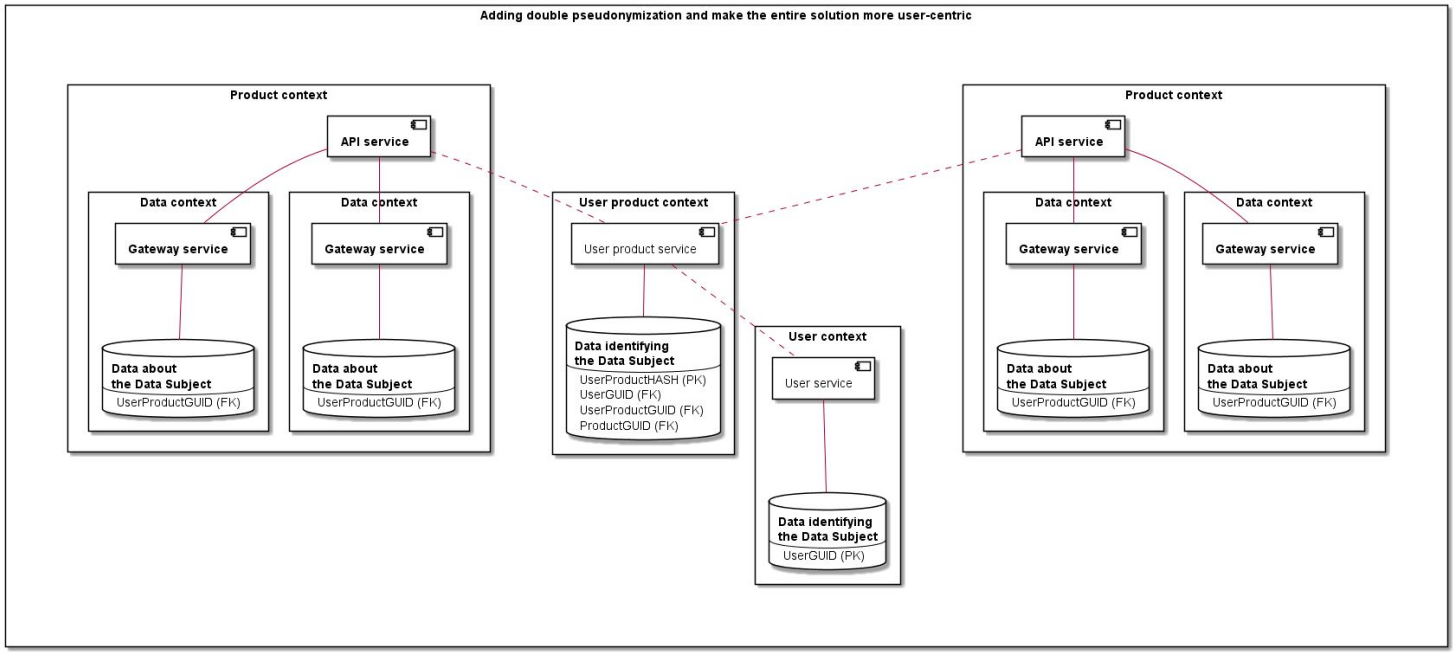
Better end-to-end security

User-centered access control: Providing a ‘user switch’ for controlling data.



Double pseudonymization pattern





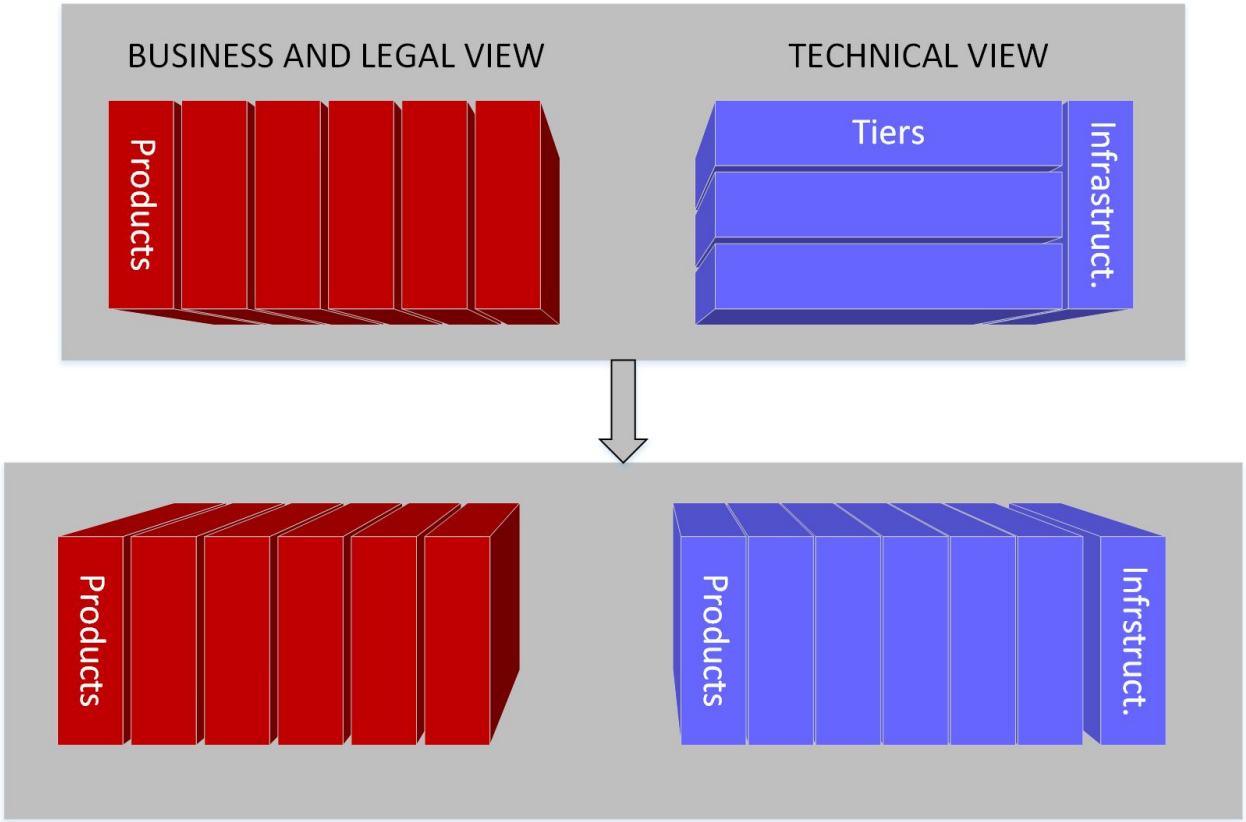
Final conclusions and perspectives

Microservices and GDPR compliance

Did applying a microservice architecture ensure GDPR compliance?

No

But the microservice architecture made lots of things easier


BUSINESS AND LEGAL VIEW
TECHNICAL VIEW
Products
Tiers
Infrastruct.
Products
Products
Infrstruct.

Did we fulfill The 7 Foundational Principles of Privacy by Design?

1. Proactive not Reactive; Preventative not Remedial
2. Privacy as the Default Setting
3. Privacy Embedded into Design
4. Full Functionality – Positive-Sum, not Zero-Sum
5. End-to-End Security – Full Lifecycle Protection
6. Visibility and Transparency – Keep it Open
7. Respect for User Privacy – Keep it User-Centric

(Source: Ann Cavoukian, Privacy by Design: The 7 Foundational Principles)

How useful were the Privacy Design Strategies?

Data oriented strategies:

1. **MINIMISE** - The amount of personal data that is processed should be restricted to the minimal amount possible.
2. **HIDE** - Any personal data, and their interrelationships, should be hidden from plain view.
3. **SEPARATE** - Personal data should be processed in a distributed fashion, in separate compartments whenever possible.
4. **AGGREGATE** - Personal data should be processed at the highest level of aggregation and with the least possible detail in which it is (still) useful.

Process oriented strategies:

1. **INFORM** - Data subjects should be adequately informed whenever personal data is processed.
2. **CONTROL** - Data subjects should be provided agency over the processing of their personal data.
3. **ENFORCE** - A privacy policy compatible with legal requirements should be in place and should be enforced.
4. **DEMONSTRATE** - Be able to demonstrate compliance with the privacy policy and any applicable legal requirements.

(Source: Jaap-Henk Hoepman)

About us...

Thomas Krogsgaard Holme

tho@sundhed.dk

Tobias Uldall-Espersen

tue@sundhed.dk