

SMART APPLICATIONS PLATFORM

DDD & Event Sourcing à la rescousse pour implémenter la GDPR ?

@JeromeAvouustin & @glours

Qui sommes nous ?



Olaf



Sven

Qui sommes-nous ?



EU General Data Protection Regulation

www.eugdpr.org

Définition



Le GDPR (ou RGPD en français, Règlement Général sur la Protection des Données) constitue le nouveau texte de référence en matière de protection des données à caractère personnel. Il renforce et unifie la protection des données pour les individus au sein de l'Union Européenne

EU General Data Protection Regulation : Qui ?

Entités concernées



Toutes les entreprises de l'UE (28 pays membres) et les entreprises mondiales traitant des données personnelles de citoyens européens

EU General Data Protection Regulation : Quand ?

Date de mise en application



25 mai 2018. Pour les traitements antérieurs à cette date, les entreprises disposeront d'un délai de deux ans pour les mettre en conformité, soit 2020. En France, en cas de non conformité, les entreprises peuvent être sanctionnées par la CNIL (jusqu'à 4% du CA de l'entreprise, plafonné à 20 millions). D'après Gartner, à peine 50% des entreprises EU seront en conformité fin 2018

EU General Data Protection Regulation : Quoi ?

Données concernées



Toute information relative à une personne physique identifiée ou qui peut être identifiée, directement ou indirectement, par référence à un numéro d'identification ou à un ou plusieurs éléments qui lui sont propres

GDPR et l'entreprise : Une transformation nécessaire



Une architecture IT hétérogène à mettre en conformité



GDPR



Nouveau poste - Data Protection Officer (DPO)
Se doit de tout documenter



Sécurité - notification des failles
(en France, 72h pour notifier la CNIL)

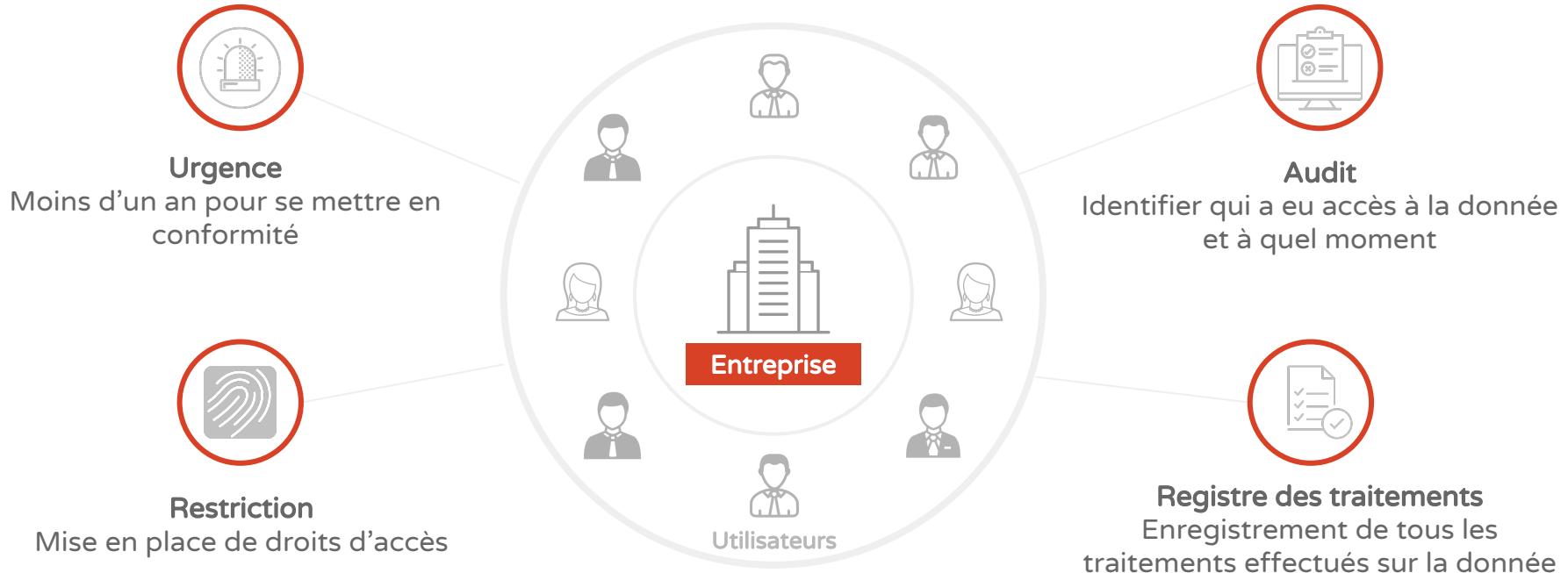


Nouvelles notions - "Privacy by design", consentement, anonymisation, droit à l'oubli, portabilité



Nouveaux processus - registre des traitements et analyse d'impact
(description systématique des opérations de traitement)

GDPR et l'entreprise : Les problématiques



GDPR et l'entreprise : Les problématiques



Anonymisation / Pseudonymisation



GDPR et l'entreprise : Les besoins



Consentement



Anonymisation



Portabilité &
Accès aux data



Habilitation



Audit / Traçabilité



Privacy by Design

Saagie Data Governance

The screenshot displays the Saagie Data Governance platform interface, featuring several key components:

- Top Navigation:** Domains (selected), Datasets, Log in.
- Search Bar:** Search a domain... and Search a dataset....
- Domain Overview:** Finance (3 datasets), Health (12 datasets), Marketing (6 datasets), Operation (8 datasets).
- Saagie Main View:** Shows Sales (7 datasets). Includes tabs for GENERAL, Datasets (selected), Provenances, Databases, Trust levels, and Data status. Sub-tabs include PERSONAL DATA, Personal data, Consent indicator, Entry date indicator, Anonymization enabled, and Anonymize.
- Dataset List:** Lists datasets such as piaf, Prospect, google, CRM events, liste, and twitter_dally, categorized by Domains (Marketing).
- Prospect Detail View:** Shows a detailed view of the Prospect dataset, including sections for Personal data, User consent field, and Entry date field.
- Analytics View:** Shows a summary of datasets containing personal data, with 33.3% having personal data. It includes sections for Personal data, Datasets containing personal data, and a summary of fields like user consent, entry date, and anonymization.
- Bottom Buttons:** SQL Query and Saagie Bot.

Saagie Data Governance: État des lieux



Saagie Data Governance: Notre analyse

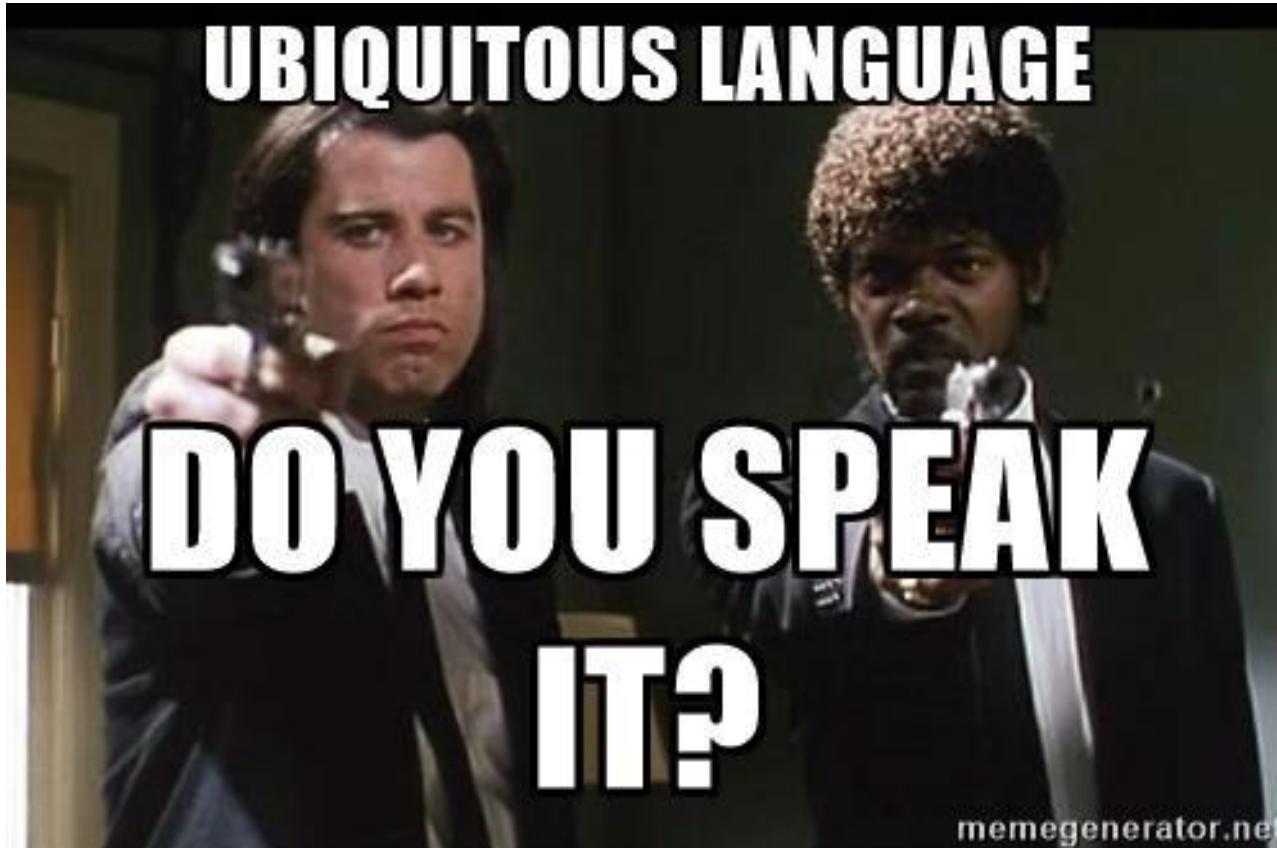


Domain Driven Design

« It's not stakeholder knowledge
but developers' ignorance
that gets deployed into production »

Alberto Brandolini - @ziobrando

Ubiquitous Language

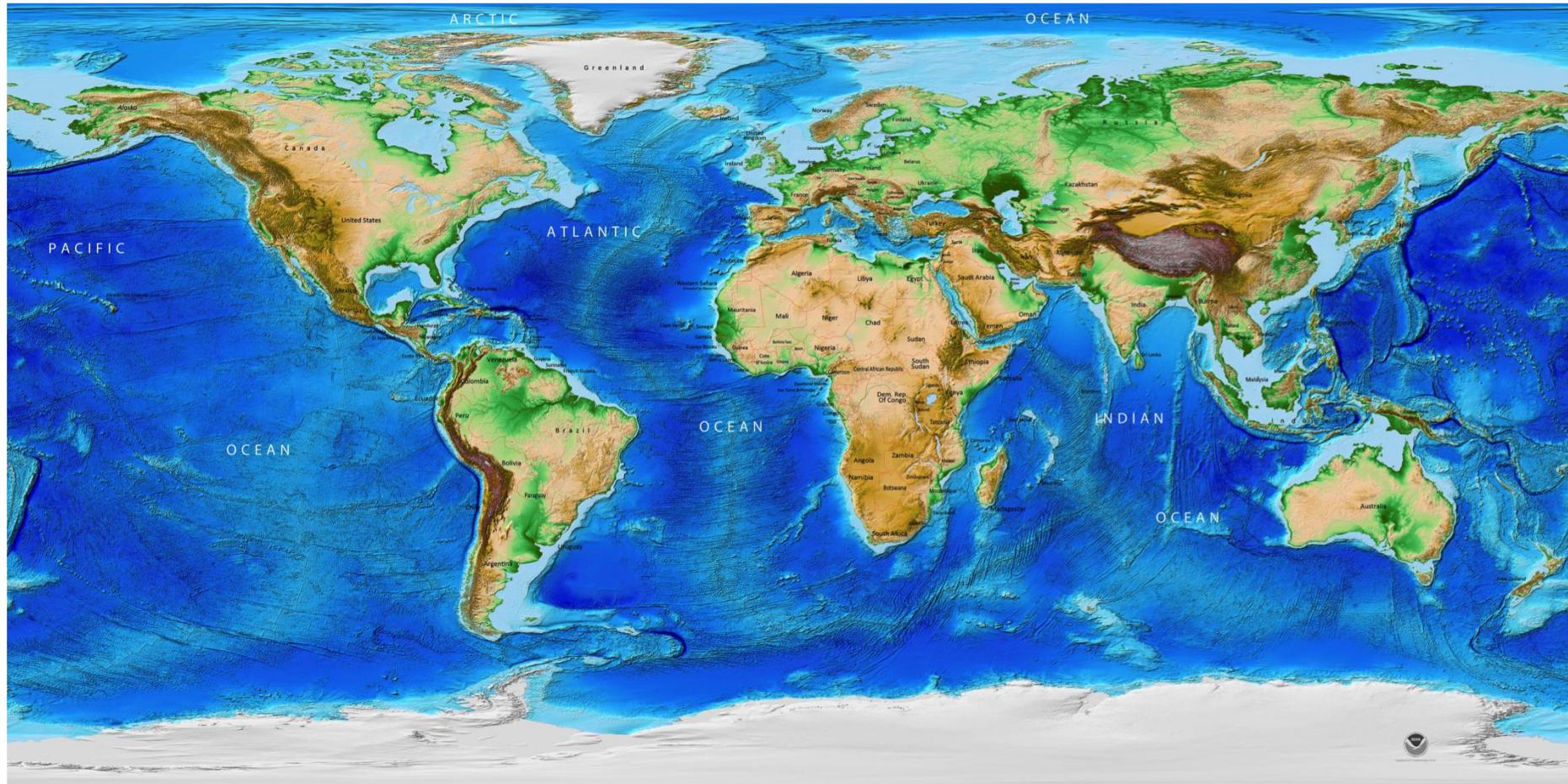


ONE MODEL



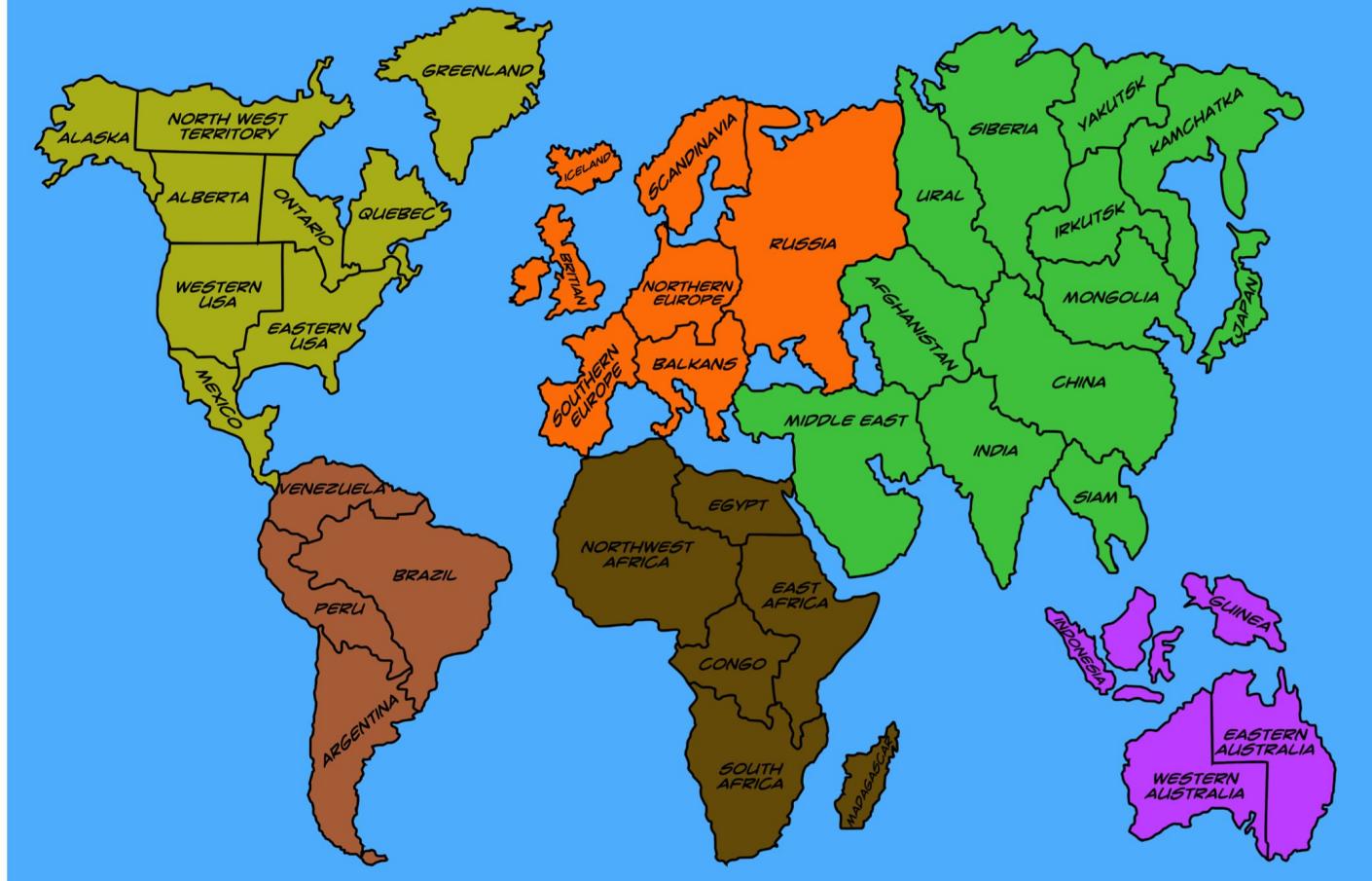
TO RULE THEM

ALL





NEWBOLA OUTBREAK - THE MATRIX GAME



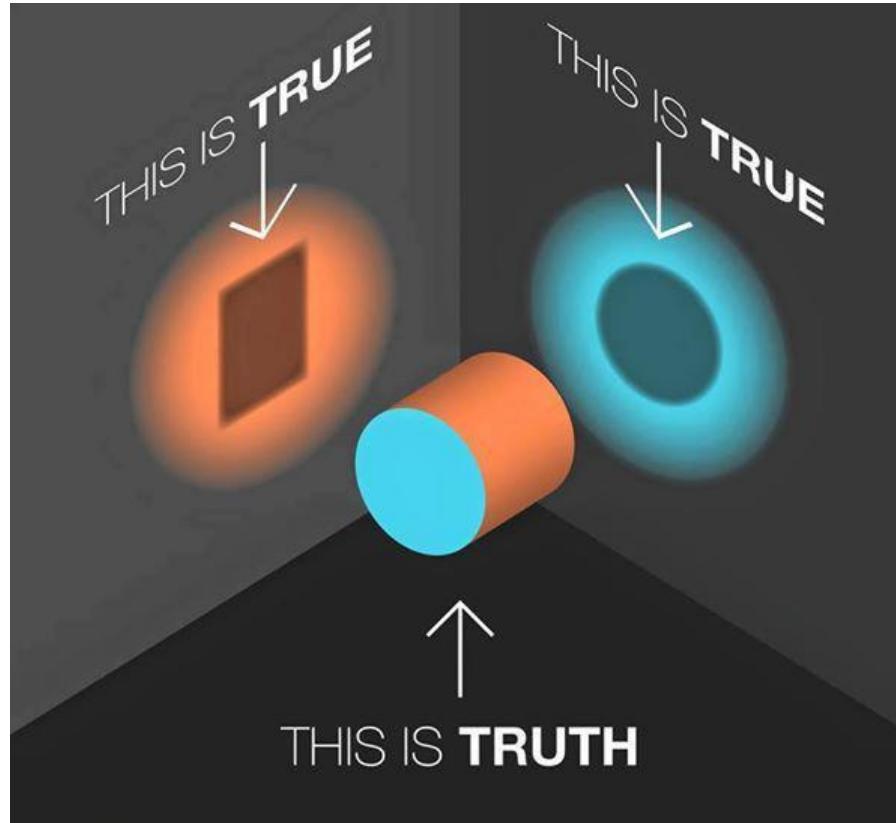
World Population Cartogram 2015

each square = 500,000 people





Bounded Contexts



Bounded Contexts



Domain Driven Design

UI

API

Services

Data access

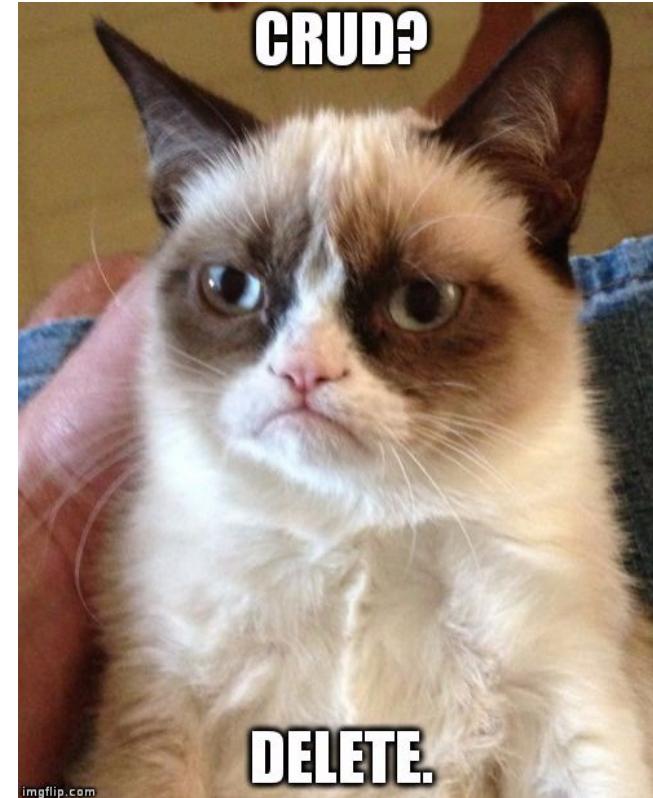


Data driven

Domain Driven Design



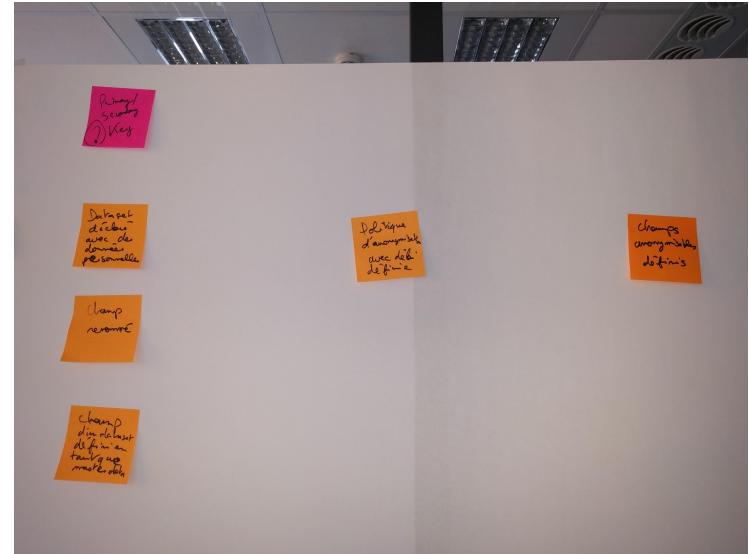
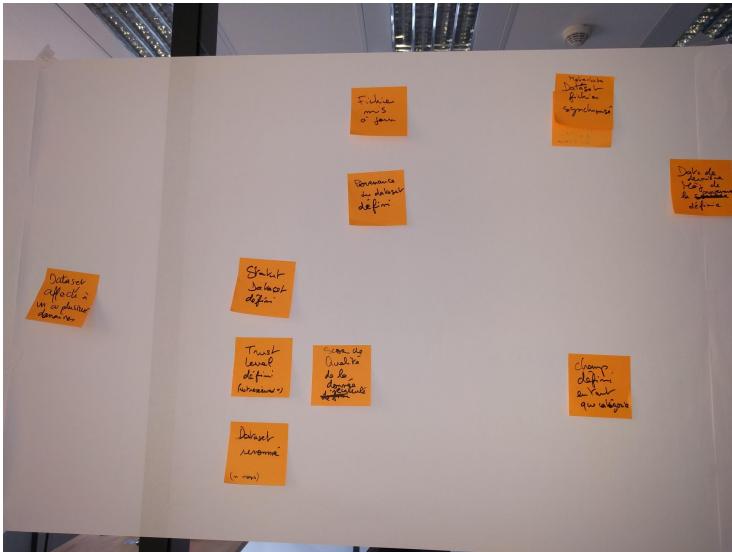
Data driven



Domain Driven Design

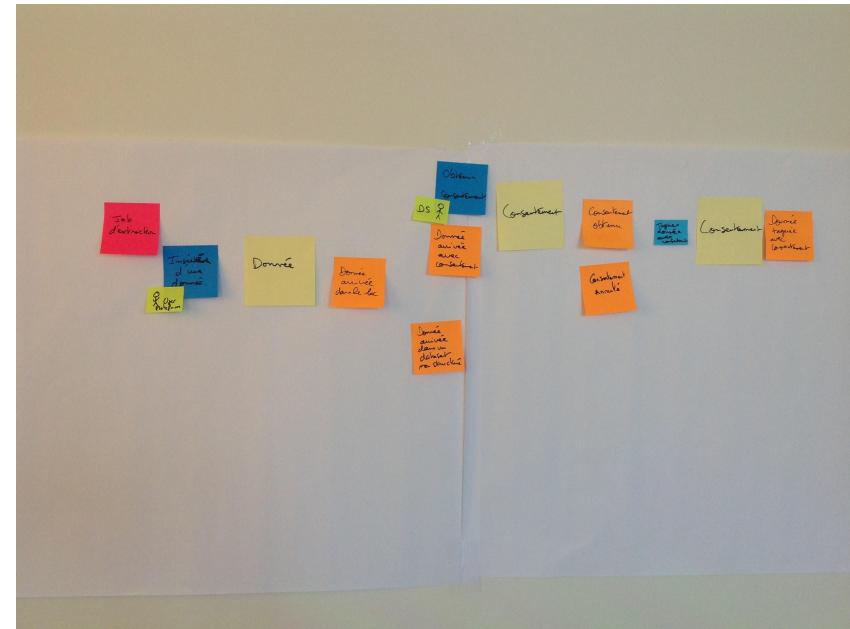
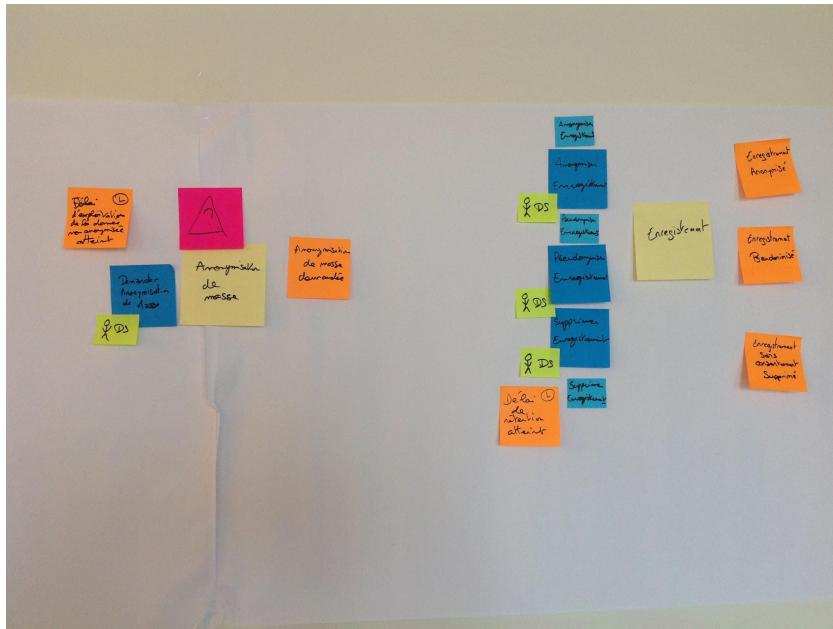


Explorer notre domaine métier : Event Storming



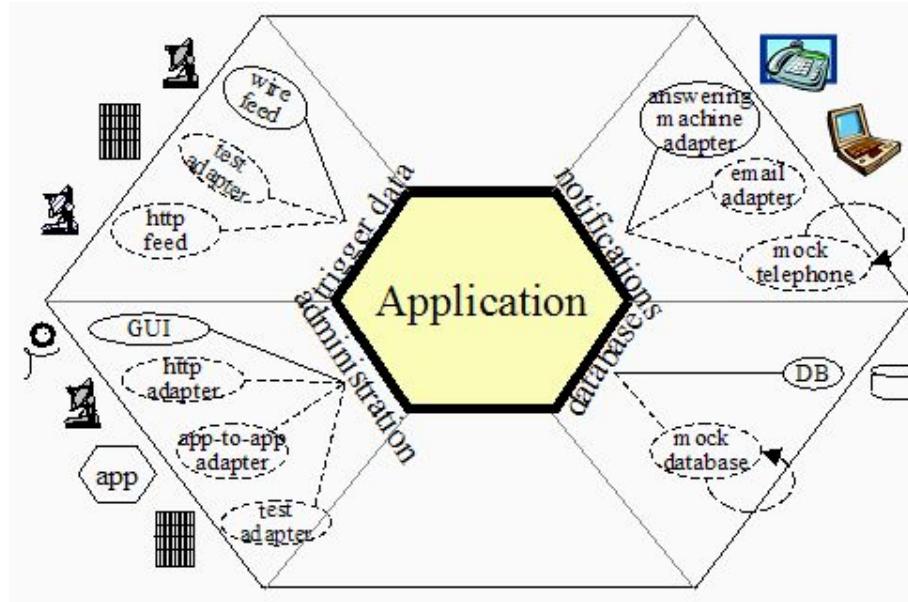
Émergence du vocabulaire commun (Ubiquitous Language) &
compréhension des besoins métiers attendus

Comprendre les implications de la GDPR : Event Storming aussi



Identification de Ubiquitous Language lié à la GDPR et des impacts sur le fonctionnement actuel de l'application

Nettoyer le core domain : Hexagonal Architecture



<http://alistair.cockburn.us/Hexagonal+architecture+complex+example.gif>

Définie par Alistair Cockburn et connu aussi sous le nom de “Ports and Adapters” ou “Object Structural”

Data Governance sous le capot

- 1 application Spring Boot
- Séparation en couches "stackées" (Controller/Service/Dao)
- Persistance dans MongoDB
- Des connecteurs sur différents composants du lac de données HDFS

Nettoyer le core domain : exemple

```
package io.saagie.datagov.model.request;
```

```
import com.fasterxml.jackson.databind.annotation.JsonSerialize;  
import io.saagie.datagov.model.request.serializer.DataRecordSerializer;
```

```
import java.util.List;  
import java.util.Optional;  
import java.util.stream.Stream;
```

```
import static java.util.stream.Collectors.toList;
```

```
@JsonSerialize(using = DataRecordSerializer.class)
```

```
public class DataRecord {  
    private final List<RecordEntry> recordEntries;
```

```
...  
}
```

```
package io.saagie.datagov.domain.data.access.user;
```

```
import java.util.List;  
import java.util.stream.Stream;
```

```
public class DataRecord {  
    private final List<RecordEntry> recordEntries;
```

```
} ...  
}
```



Nettoyer le core domain : sans tout casser

```
@Configuration  
public class WebConfiguration extends WebMvcConfigurerAdapter {  
  
    @Bean  
    public MappingJackson2HttpMessageConverter mappingJackson2HttpMessageConverter() {  
        ObjectMapper mapper = new ObjectMapper();  
        mapper.registerModule(new JavaTimeModule());  
        mapper.registerModule(new VavrModule());  
        mapper.registerModule(dataRecordModule());  
  
        return new MappingJackson2HttpMessageConverter(mapper);  
    }  
  
    private Module dataRecordModule() {  
        SimpleModule module = new SimpleModule();  
        module.addSerializer(DataRecord.class, new DataRecordSerializer());  
        return module;  
    }  
}
```

Nettoyer le core domain : les services aussi !

```
package io.saagie.datagov.service;

import io.saagie.datagov.dao.ImpalaDao;
import io.saagie.datagov.dao.SaagieBotDao;
import io.saagie.datagov.model.request.CompactDataset;
import io.saagie.datagov.model.request.RequestResult;
import io.saagie.datagov.model.request.SaagieBotResponse;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

@Service
public class RequestService {

    @Autowired
    private ImpalaDao impalaDao;

    @Autowired
    private SaagieBotDao saagieBotDao;

    public RequestResult sampleRequest(String query) {
        return this.impalaDao.testQuery(query);
    }
    ...
}
```

Nettoyer le core domain : Service sans dépendance

```
package io.saagie.datagov.domain.data.access.user;

import io.saagie.datagov.model.request.CompactDataset;

public class QueryService {

    private DataQuery dataQuery;

    public QueryService(DataQuery dataQuery) {
        this.dataQuery = dataQuery;
    }

    public QueryResult sampleQuery(final String query) {
        return this.dataQuery.sampleQuery(query);
    }

    ...
}
```

Pattern Repository

```
package io.saagie.datagov.domain.data.access.user;

import io.saagie.datagov.model.request.CompactDataset;

public interface DataQuery {

    QueryResult sampleQuery(final String query);

    QueryResult executeQuery(final String query);

    CompactDataset executeQueryAsCompactDataset(final String query);

    boolean isSaagieBotAvailable();

    SaagieBotResponse executeBotQuery(final String query);
}
```

```
package io.saagie.datagov.infra.adapter;
```

...

```
@Repository  
public class DataQueryRepository implements DataQuery {
```

```
    @Autowired  
    private SaagieBotDao saagieBotDao;
```

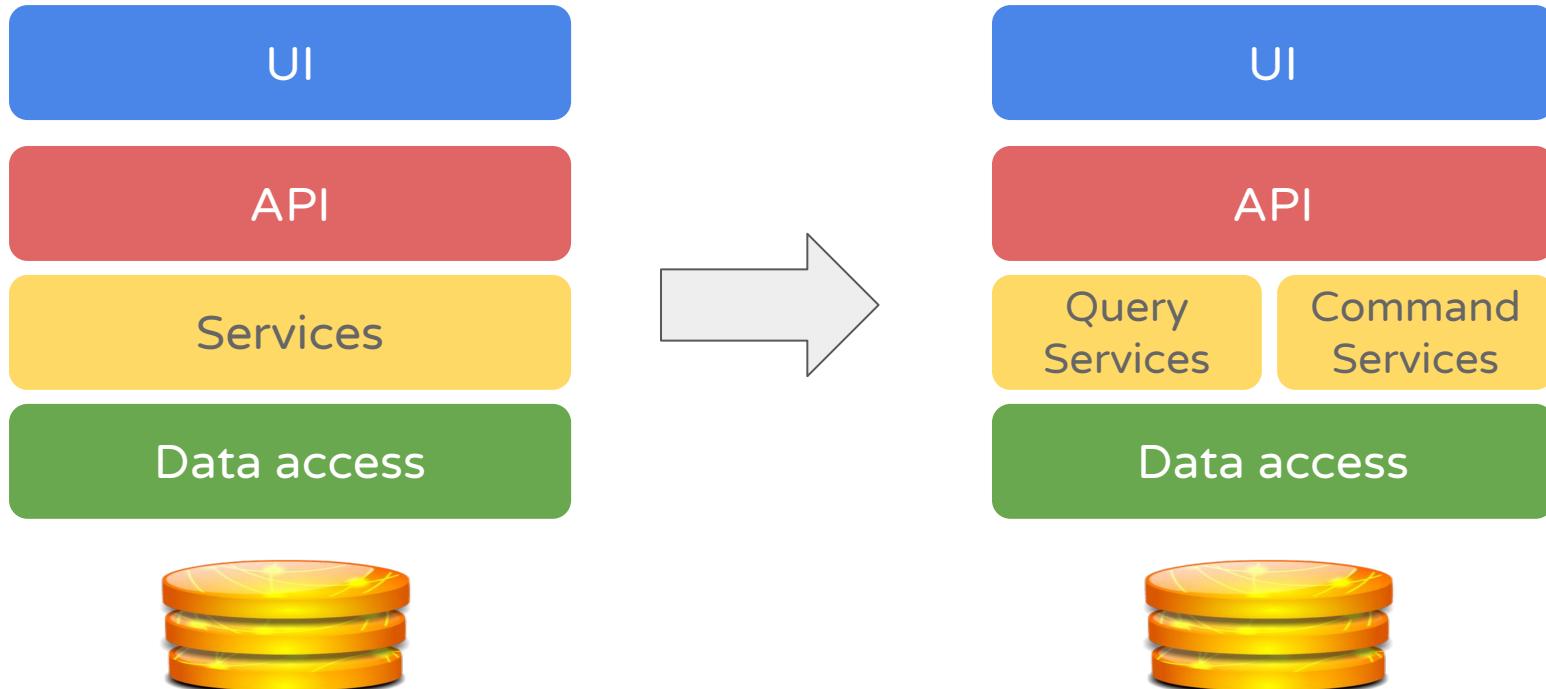
```
    @Autowired  
    private QueryDao queryDao;
```

```
    @Override  
    public QueryResult sampleQuery(String query) {  
        return this.queryDao.testQuery(query);  
    }
```

```
    @Override  
    public QueryResult executeQuery(String query) {  
        return this.queryDao.executeQuery(query);  
    }
```

```
    @Override  
    public CompactDataset executeQueryAsCompactDataset(String query) {  
        return this.queryDao.executeQueryToCompactDataset(query);  
    }  
    ...  
}
```

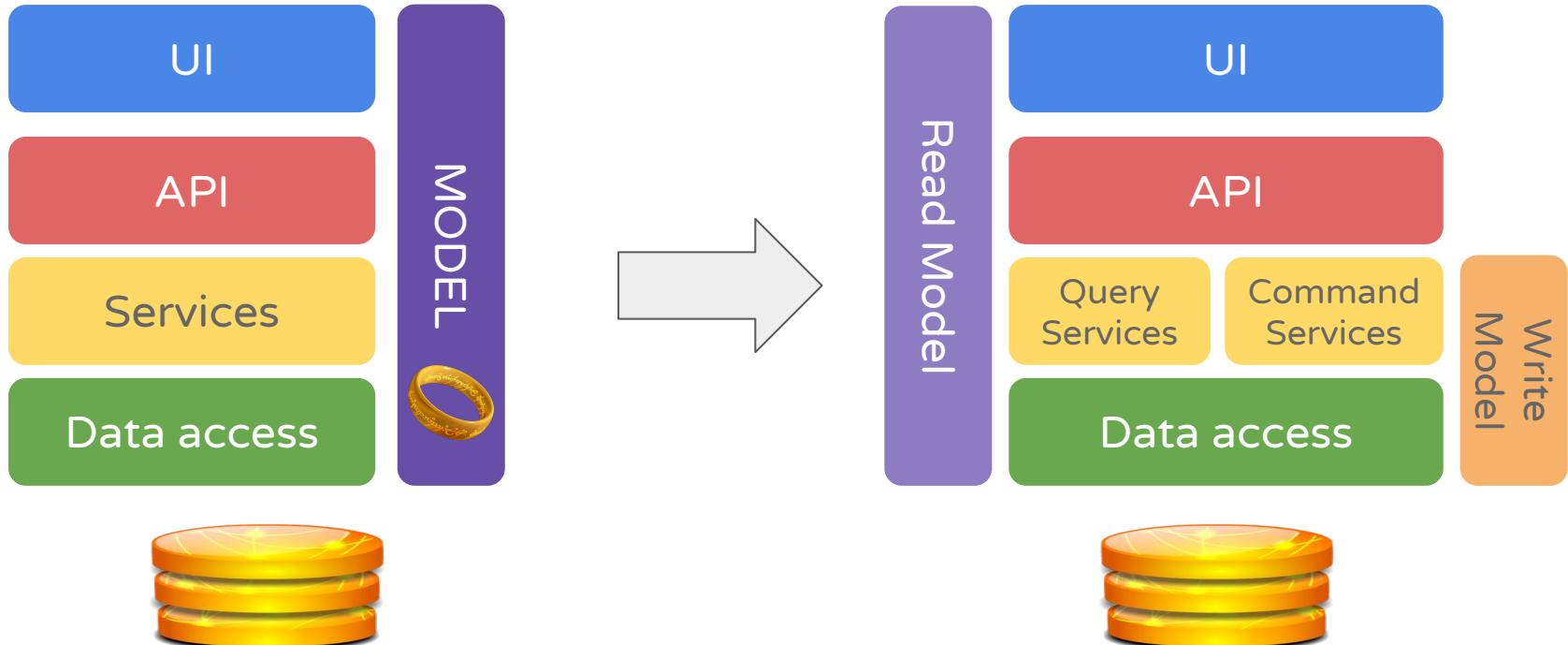
Command Query Responsibility Segregation



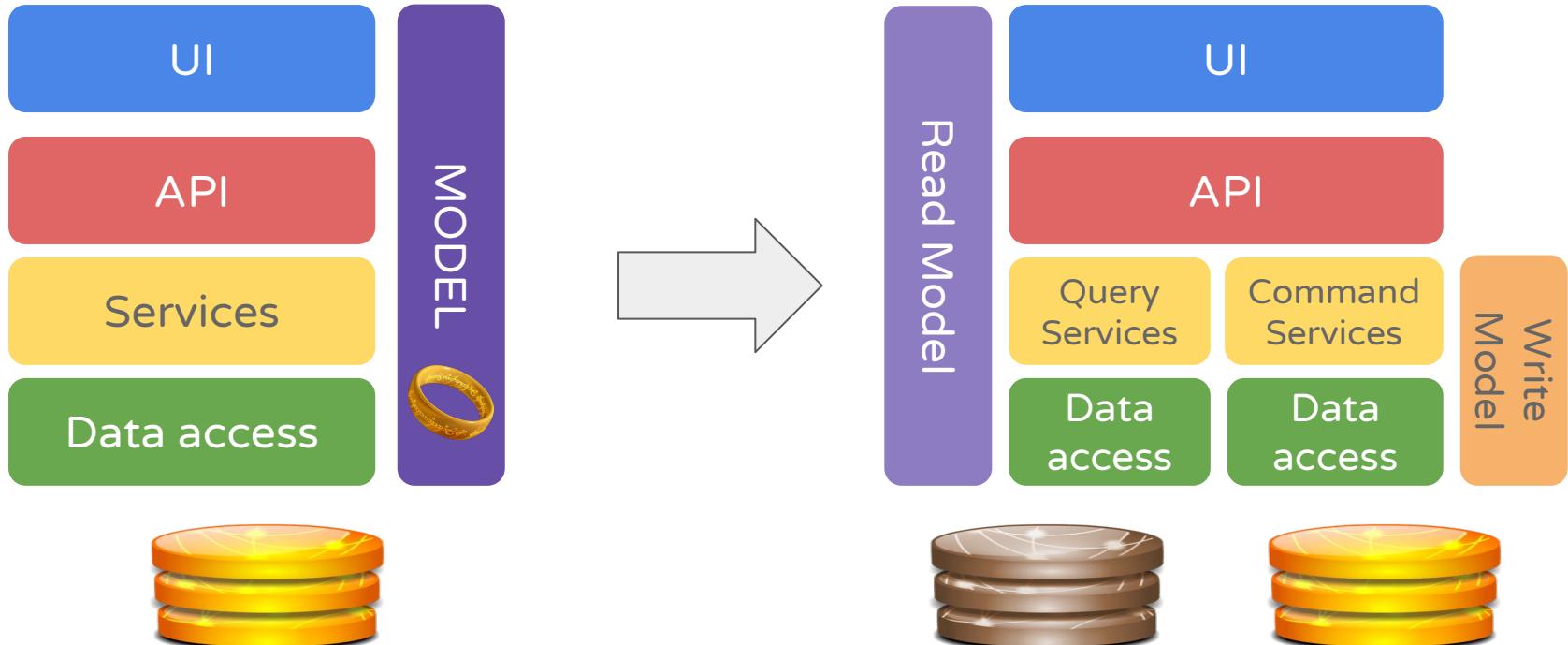
Command Query Responsibility Segregation



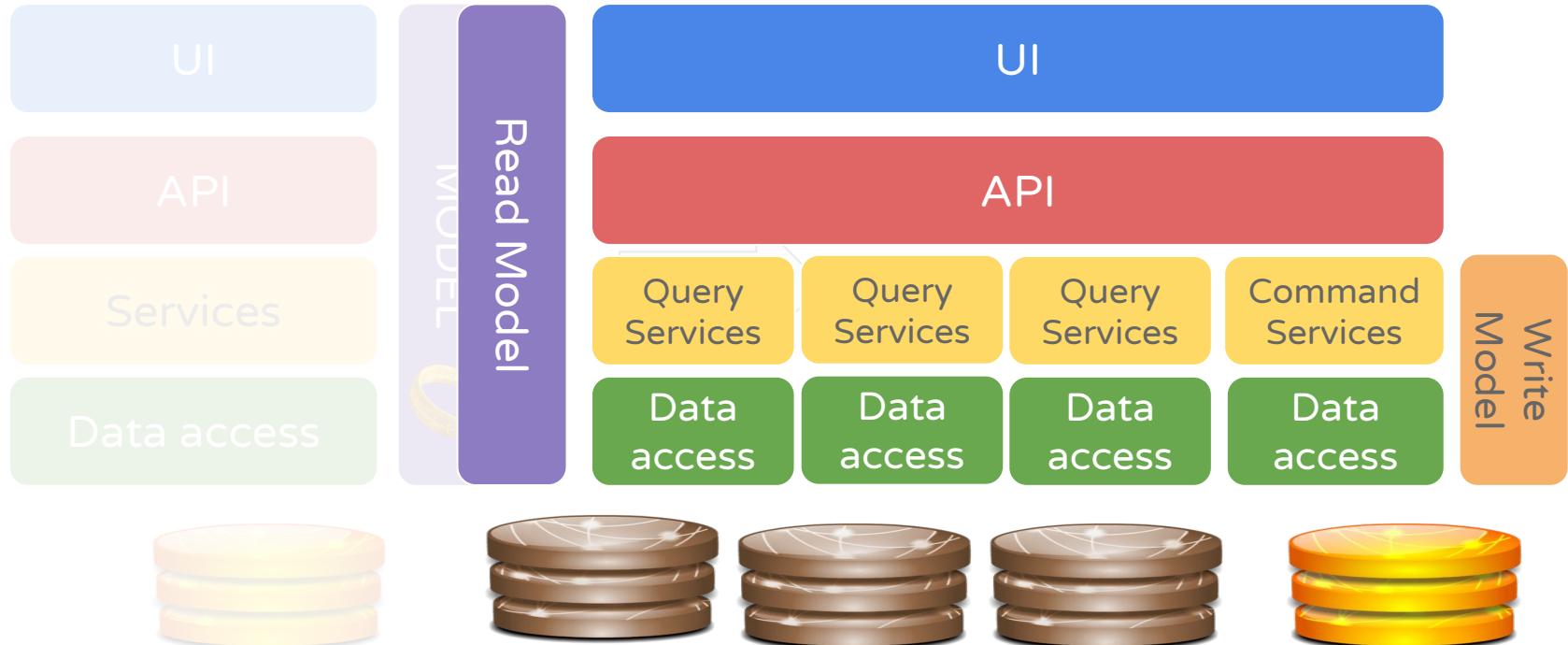
Command Query Responsibility Segregation



Command Query Responsibility Segregation



Command Query Responsibility Segregation



Event Sourcing

- Persistance de l'historique des événements
- Les événements deviennent la source de vérité, pour reconstituer l'état
- Source de données immutable (append only)

Event Sourcing

idPotAuFeu	nbCarottes	nbChoux	etat
42	0	0	en prépa
43	2	3	en cuisson
44	5	3	prêt

Event Sourcing

Commande "Ajouter une carotte au pot au feu 42"

idPotAuFeu	nbCarottes	nbChoux	etat
42	0	0	en prépa
43	2	3	en cuisson
44	5	3	prêt

Event Sourcing

Commande "Ajouter une carotte au pot au feu 42"

idPotAuFeu	nbCarottes	nbChoux	etat
42	1	0	en prépa
43	2	3	en cuisson
44	5	3	prêt

Event Sourcing

Commande "Ajouter 2 choux au pot au feu **42**"

idPotAuFeu	nbCarottes	nbChoux	etat
42	1	2	en prépa
43	2	3	en cuisson
44	5	3	prêt

Event Sourcing

Commande "Ajouter une carotte au pot au feu 42"

idPotAuFeu	nbCarottes	nbChoux	etat
42	2	2	en prépa
43	2	3	en cuisson
44	5	3	prêt

Event Sourcing

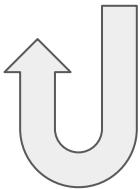
Commande "*Retirer une carotte au pot au feu 42*"

idPotAuFeu	nbCarottes	nbChoux	etat
42	1	2	en prépa
43	2	3	en cuisson
44	5	3	prêt

Event Sourcing

Commande "Ajouter une carotte au pot au feu 42"

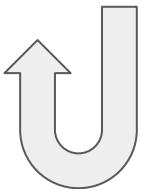
Aggrégat	Id	Event	Data	Version
PotAuFeu	42	prépa_lancée	null	1



Event Sourcing

Commande "Ajouter 2 choux au pot au feu 42"

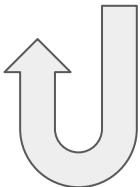
Aggrégat	Id	Event	Data	Version
PotAuFeu	42	prépa_lancée	null	1
PotAuFeu	42	carotte_ajoutée	{ nb : 1 }	2



Event Sourcing

Commande "Ajouter 1 carotte au pot au feu 42"

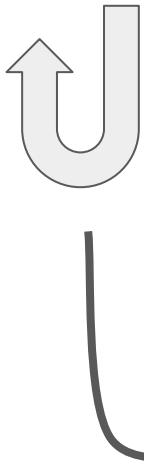
Aggrégat	Id	Event	Data	Version
PotAuFeu	42	prépa_lancée	null	1
PotAuFeu	42	carotte_ajoutée	{ nb : 1 }	2
PotAuFeu	42	chou_ajouté	{ nb : 2 }	3



Event Sourcing

Commande "*Retirer 1 carotte au pot au feu 42*"

Aggrégat	Id	Event	Data	Version
PotAuFeu	42	prépa_lancée	null	1
PotAuFeu	42	carotte_ajoutée	{ nb : 1 }	2
PotAuFeu	42	chou_ajouté	{ nb : 2 }	3
PotAuFeu	42	carotte_ajoutée	{ nb : 1 }	4



Event Sourcing

Aggrégat	Id	Event	Data	Version
PotAuFeu	42	prépa_lancée	null	1
PotAuFeu	42	carotte_ajoutée	{ nb : 1 }	2
PotAuFeu	42	chou_ajouté	{ nb : 2 }	3
PotAuFeu	42	carotte_ajoutée	{ nb : 1 }	4
PotAuFeu	42	carotte_retirée	{ nb : 1 }	5



Event Sourcing

- Testabilité (immutabilité) !
- Versioning d'aggrégats !!
- Debugging et reproductibilité !!!
- Rollback facilité !!!!
 - Et non ! pas en supprimant le dernier event !
- Traçabilité gratos !!!!!
- Recomposer des projections à la demande !!!!!!

Event Sourcing everywhere ?

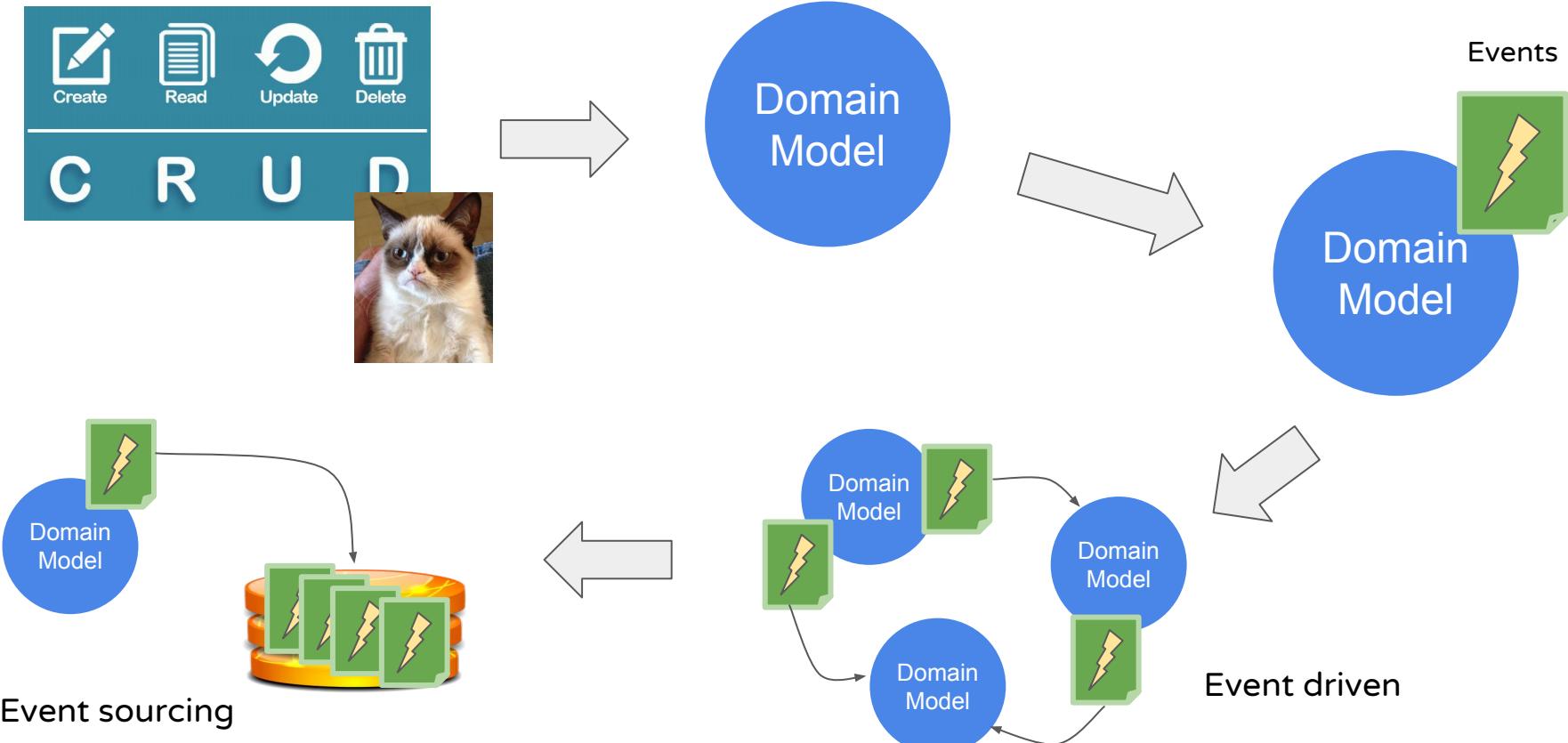
EVENT SOURCING



Event Sourcing everywhere ?



Road to Event Sourcing



GDPR : ce qu'il faut retenir

Consentement

Gérer un champ de consentement ainsi que la date d'obtention de celle-ci

Record Life

Date d'enregistrement
Date de modification + personne/job ayant modifié

Data Lineage

Tracer l'accès et la modification à la donnée idéalement connaître les delta de modification

Anonymisation

Proposer à minima un processus d'anonymisation et un processus de pseudonymisation

Où nous en sommes



Questions ?



Clear your mind of questions.