

DomainModel.stream()



DOMAINMODEL.STREAM()

<https://github.com/lesfurets/model-map>

WELCOME TO THE FURETS !

@DIGUGLI – GILLES DI GUGLIELMO

- Designer of sweet cooked software since 1999
- Software Architect at LesFurets.com

@JUBAUDRY – JULIEN BAUDRY

- Java Developer since 2007
- Software Architect at LesFurets.com

 Auto

 Santé

 Moto

 Habitation

 Finances

 Forfaits

 Cadeaux futés

 Auto

 Santé

Votre demande concerne

Un véhicule possédé actuellement ▾

Je compare >



Recevez votre cadeau futé

Recevez gratuitement votre peluche pour tout achat d'une assurance auto via LesFurets.com ***

[Comment recevoir mon cadeau ? >](#)

ÉTAPE 1

JE COMPARE

DES CENTAINES
D'OFFRES
D'ASSURANCE

ÉTAPE 2

J'ACHÈTE

UNE ÉCONOMIE
MOYENNE DE
220 €/AN*

ÉTAPE 3

JE DEMANDE
MON CADEAU



1

Votre habitation

2

Vos besoins

3

Vos coordonnées

4

Vos offres

5

Souscrire

1. Mon habitation

Ce logement est :



Un appartement



Une maison individuelle

L'habitation la plus proche se situe : 

A moins de 50 m 

Pour ce logement, je suis : 

Locataire

Votre avis

J'habite déjà ce logement ? 



Oui



Non

J'occupe ce logement comme résidence : 



Principale



Secondaire

Ce logement reste inhabité (en jours) : 





Votre véhicule



Les conducteurs



Votre assurance

4

5

Vos offres

Souscrire

Assistance

Assistance accident (8)

Assistance panne 0 km (3)

Indemnisation

Protection du conducteur (8)

Bris de glace (8)

Budget

Pas de frais de dossier (8)

Pas d'avance de frais en garage agréé (8)

Souscription

Assuré dans l'heure (2)

Souscription en ligne (8)

[Modifier les détails](#)



CITROEN ACADIANE

Formule : [Tous risques](#)



Groupe Aviva

235 € / an

Frais de dossier inclus

25 km

Assistance
panne

250 €

Franchise tout
accident

Poursuivre sur
le site de l'assureur



[Être rappelé par un conseiller](#)

[Devis par email](#)

Votre avis

nous vous
assurons !

Offre Thelem

236 € / an

Frais de dossier inclus

0 km

Assistance
panne

240 €

Franchise tout
accident

Poursuivre sur
le site de l'assureur



[Être rappelé par un conseiller](#)

[Devis par email](#)



Groupe Allianz

248 € / an

Frais de dossier inclus

50 km

Assistance
panne

240 €

Franchise tout
accident

Poursuivre sur
le site de l'assureur



[Être rappelé par un conseiller](#)

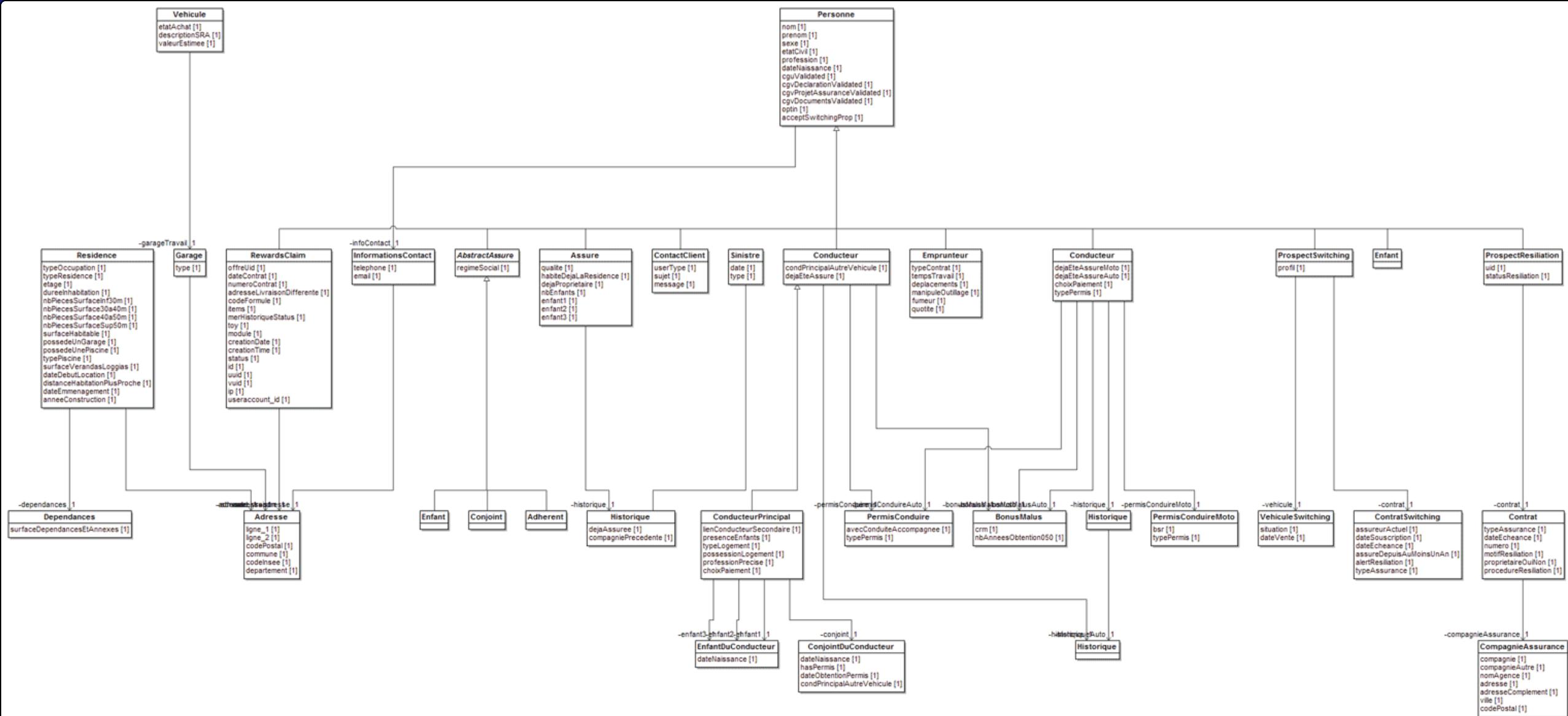
[Devis par email](#)

PROMO 2 mois offerts jusqu'au 30/11. Voir conditions sur site.



- 1 website, 5 Insurance Products : Car, Health, Home, Bike, Loan
- 1 codebase, 450k lines of code, 60k unit tests, 150 selenium tests
- 22 Developers, 2 DevOps, 3 Architects
- 10+ production servers
- 1 release per day
- 7 years of code history
- 2.5M quotes/year, 31% of market share

DOMAIN MODEL, WHAT DO YOU MEAN ?



MOVING THE ARCHITECTURE

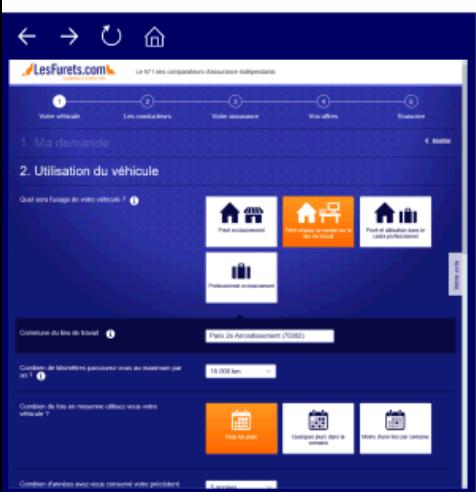
ARCHITECTURE BUZZ

Copy on write
Immutable data
NoSQL
Clustered Data Base
Data streaming
Micro batching
Lambda architecture
Data on demand

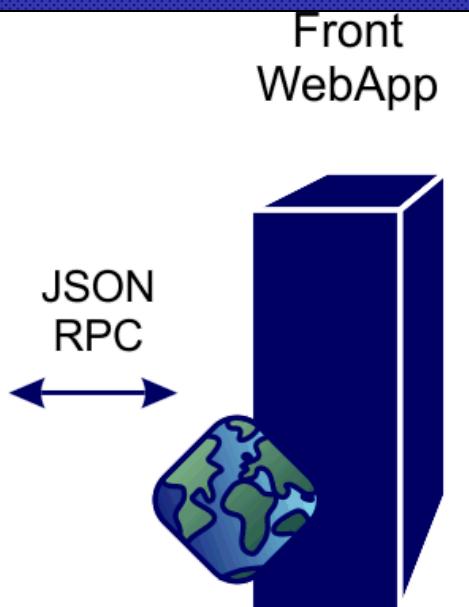


WHAT YOU HAVE ...

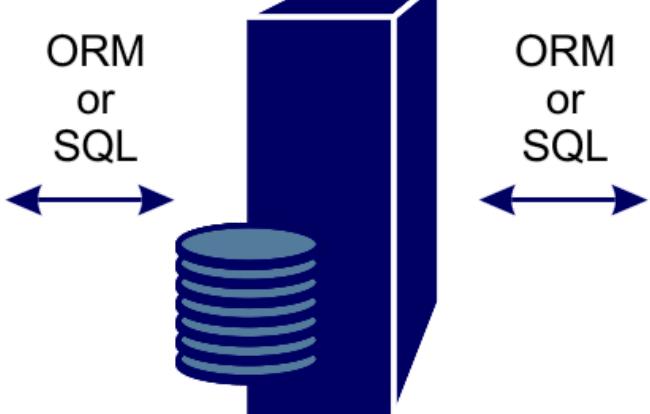
JavaScript
Front App



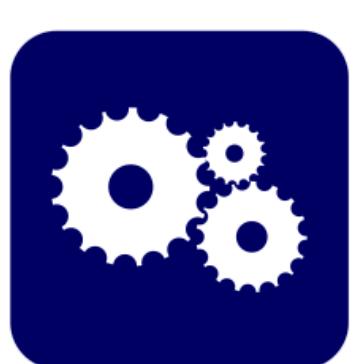
Front
WebApp



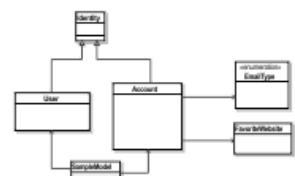
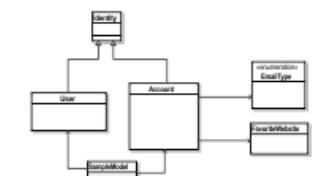
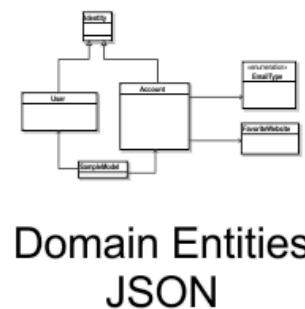
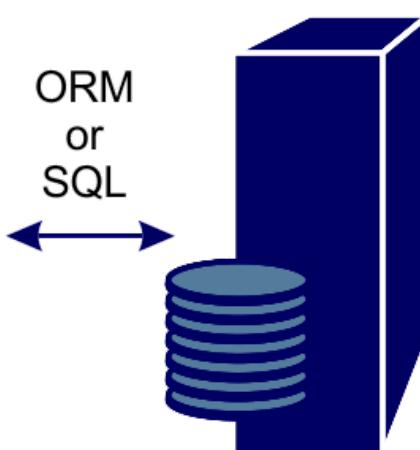
Runtime
DataBase



Batch
ETL

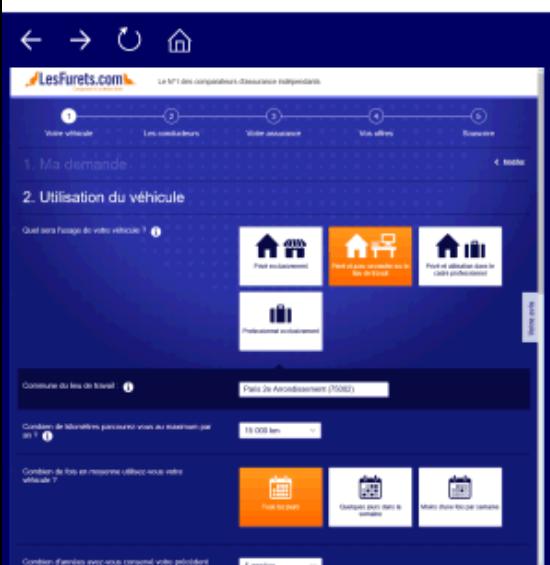


BackOffice
DataBase



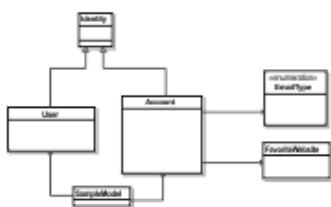
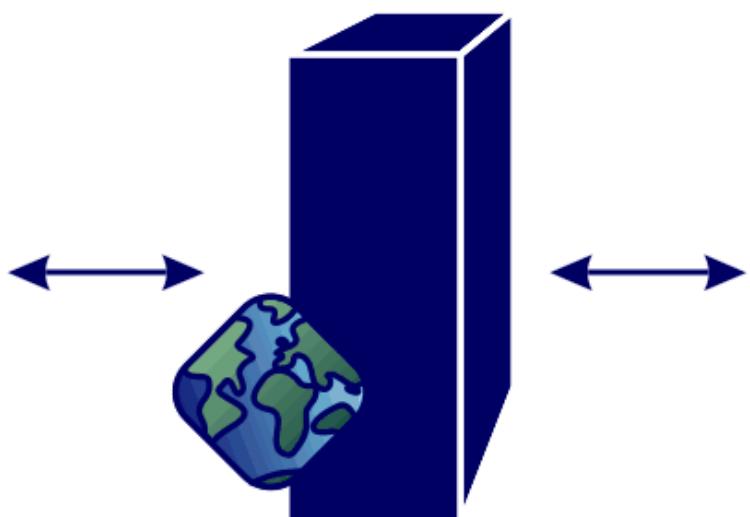
WHAT YOU WANT ...

JavaScript
Front App



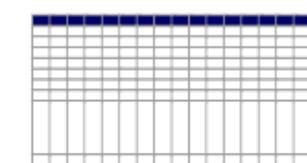
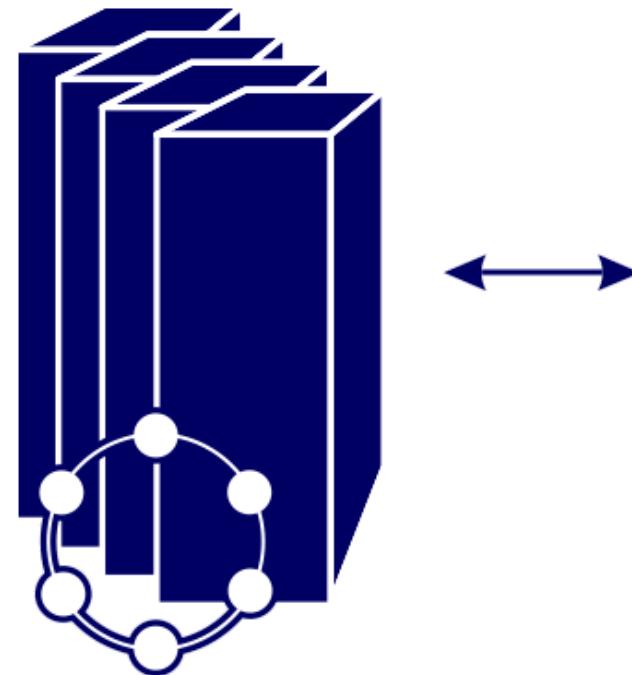
Key/Value
JSON
Dictionary

Front
WebApp



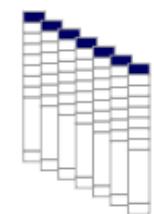
Domain Entities
Java Beans

Cassandra
Cluster



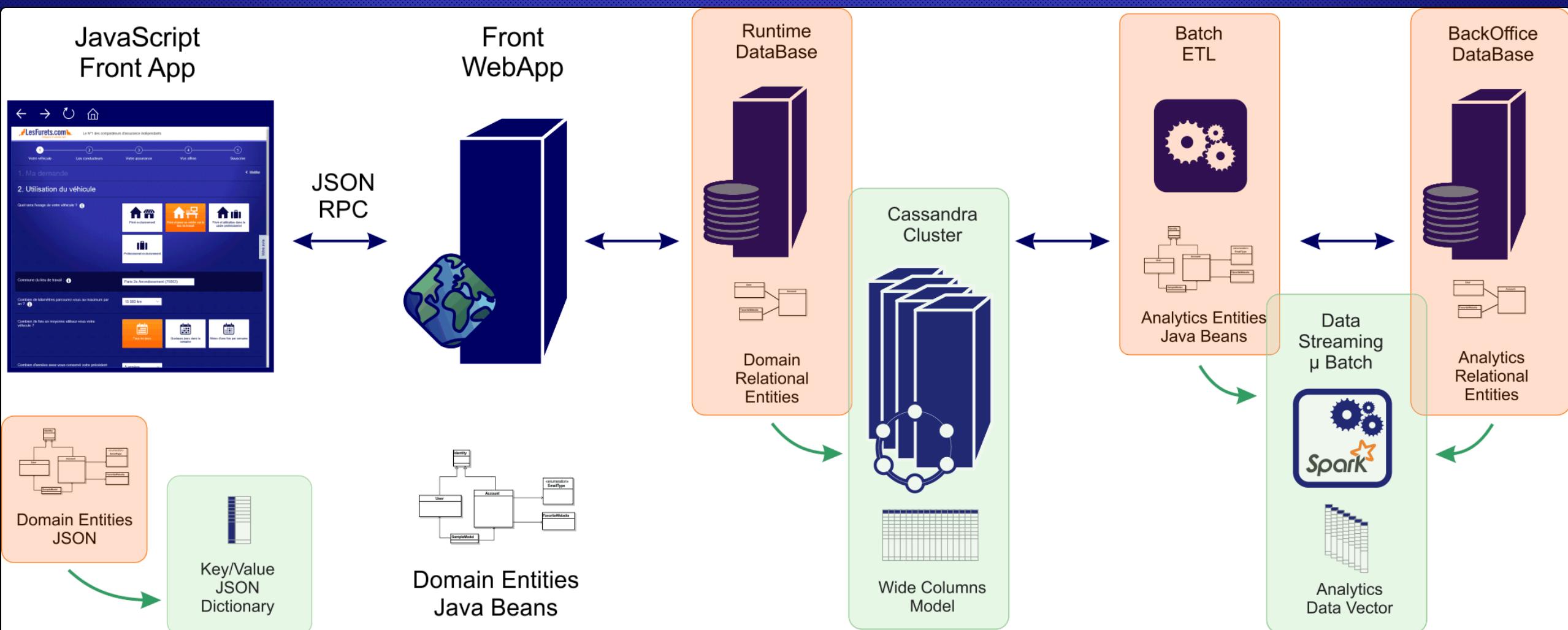
Wide Columns
Model

Data
Streaming
μ Batch



Analytics
Data Vector

YOU ARE IN TROUBLE ...



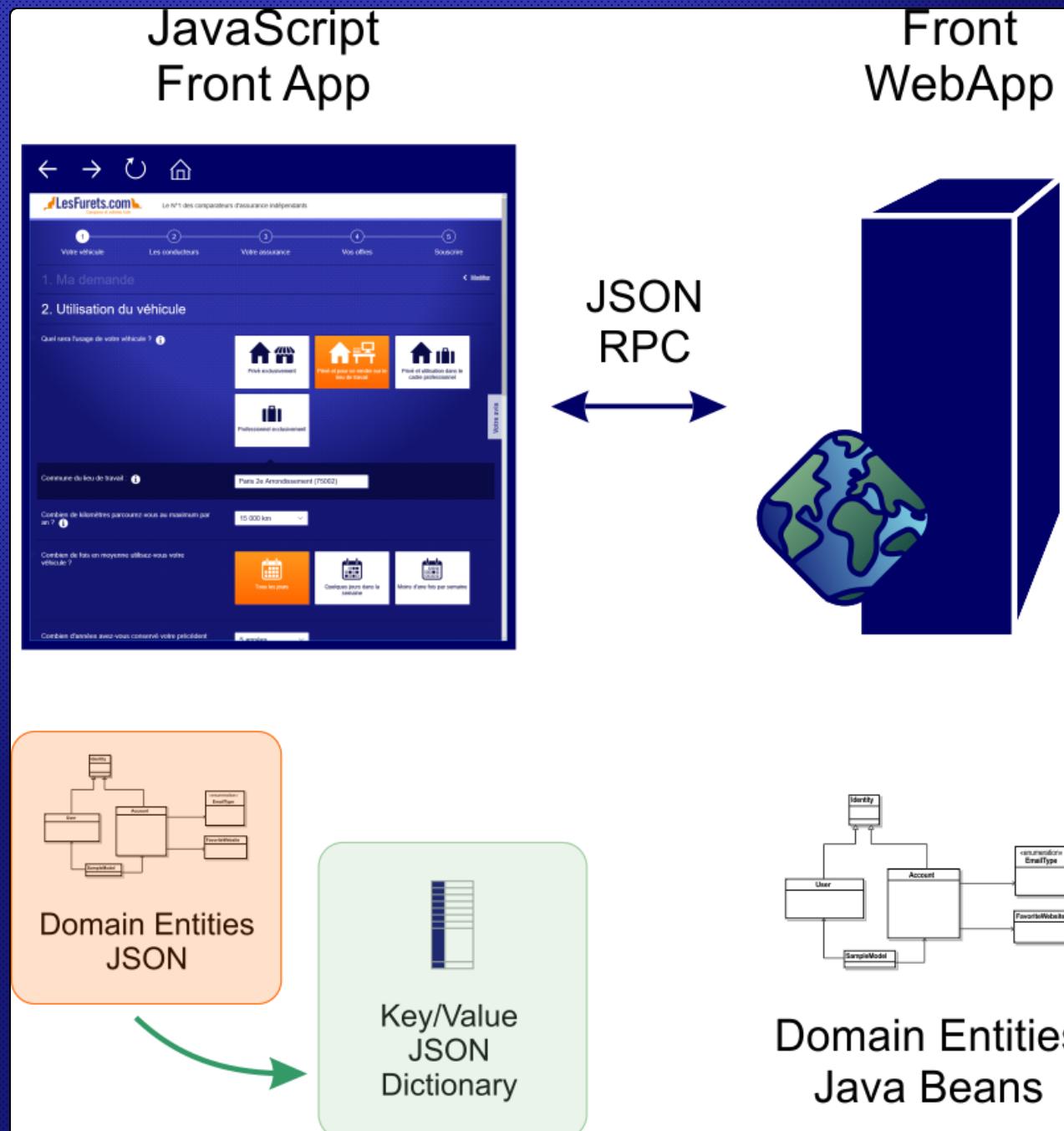
FRONT APP EVOLUTIONS

Your domain model is cloned using two different languages.

Governance of the domain model and change requests are difficult to implement.

Development owners of the server and front application are different.

UI Spaghetti code.



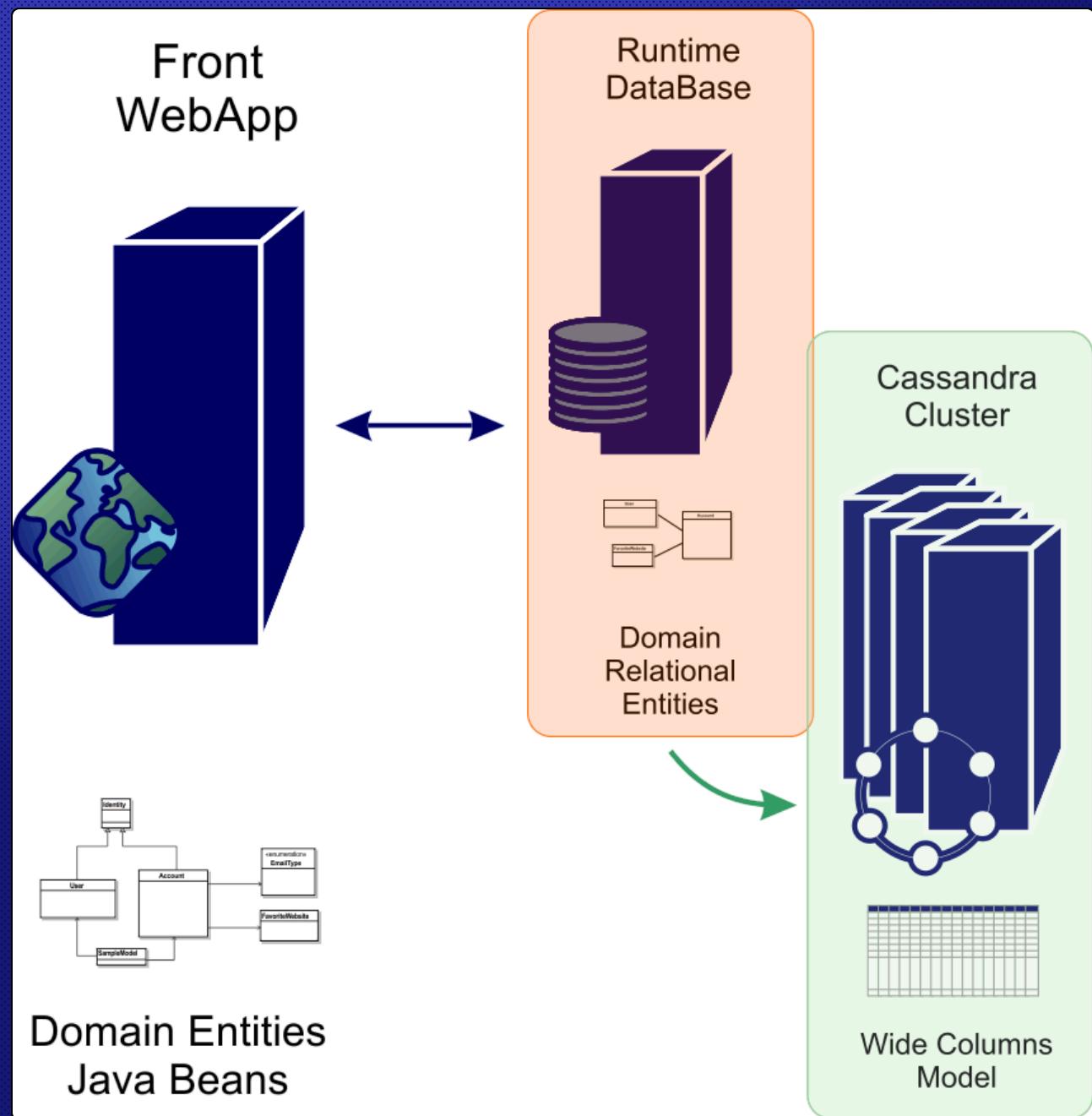
FRONT STORAGE EVOLUTIONS

Mapping entities in Cassandra is painful.

Don't migrate your Entity/Relation model using cut & paste!

You should probably flat your domain model to fit a single row.

Infinite storage.



BACKOFFICE EVOLUTIONS

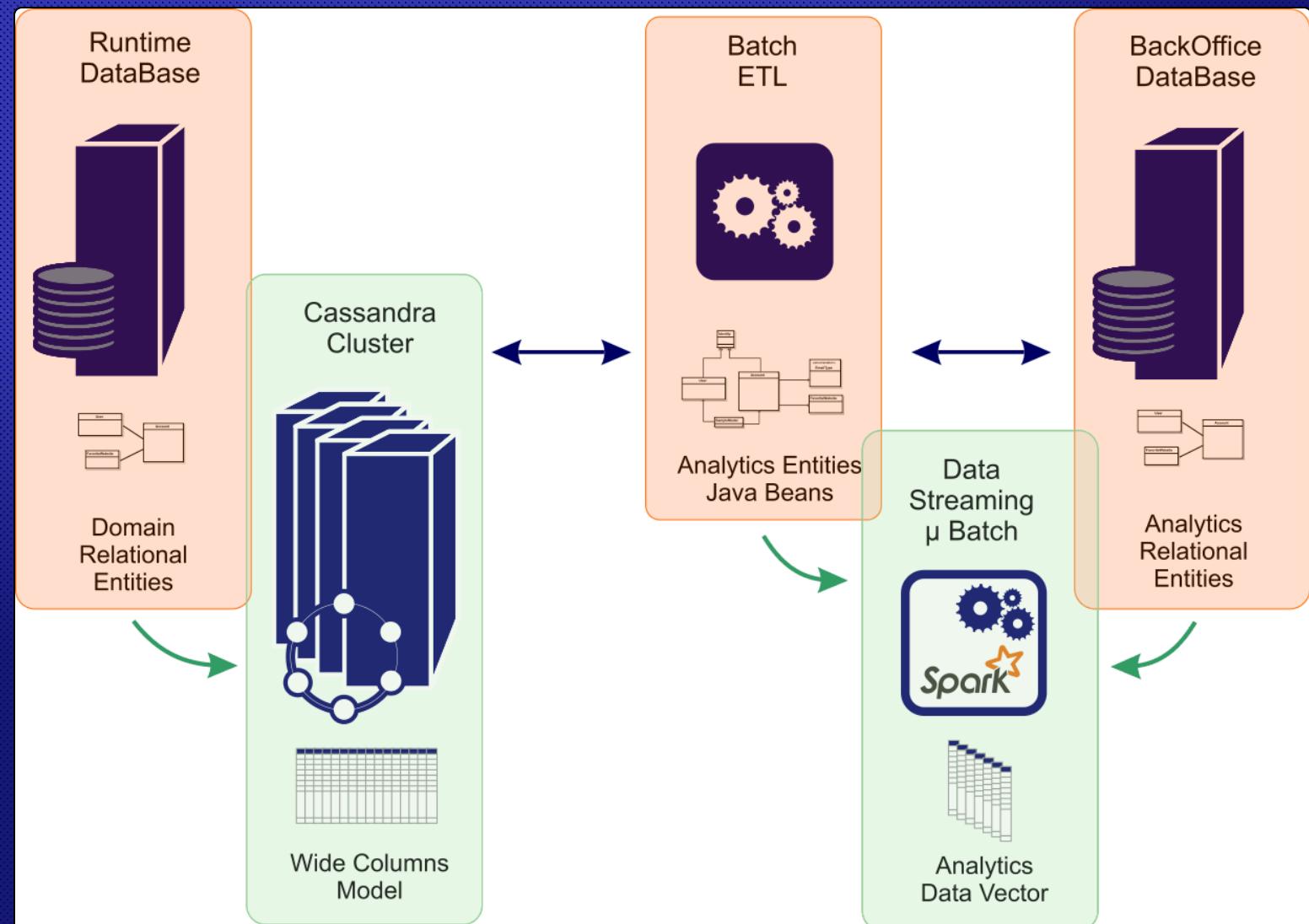
A lot of rework to implement.

Spark and ML-lib love data vectors.

Spark read raw data, aggregate, write them in microbatches.

No more (very long) nightly batches.

Full history is available for custom computations.



WHAT WE DO NOW ?

My Informations

My Settings and Email

Language English Français

I accept to receive email Yes No

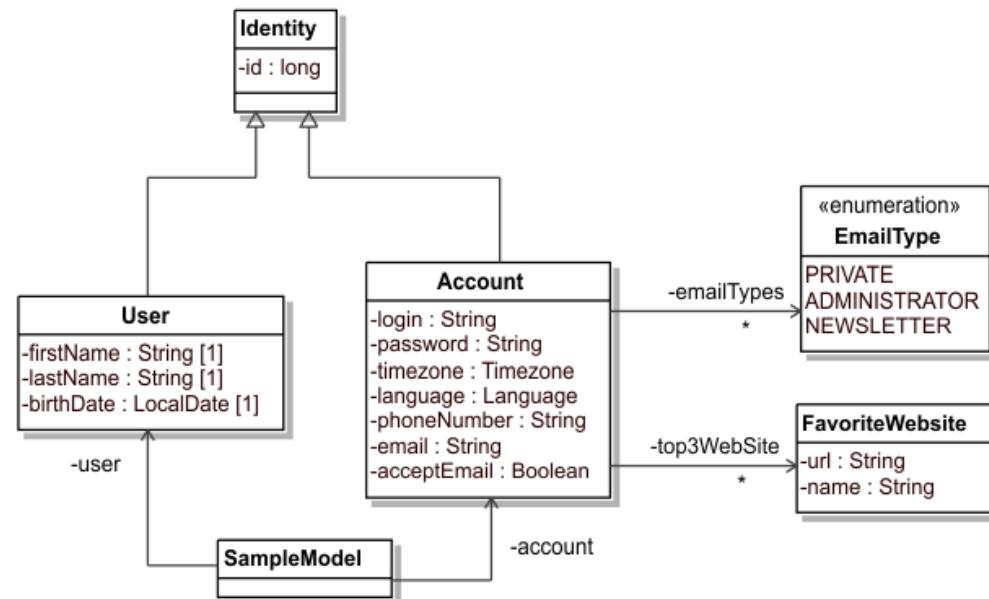
Email preference Private messages Administrator Newsletter

Emails limit / week 0

Timezone GMT+00:00

Submit

My Account



ADD KEYS EVERYWHERE !

My Informations

My Settings and Email

Language English Français

I accept to receive email Yes No

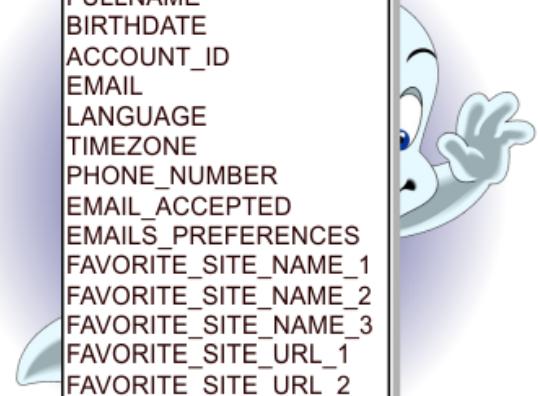
Email preference Private messages Administrator Newsletter

Emails limit / week 0

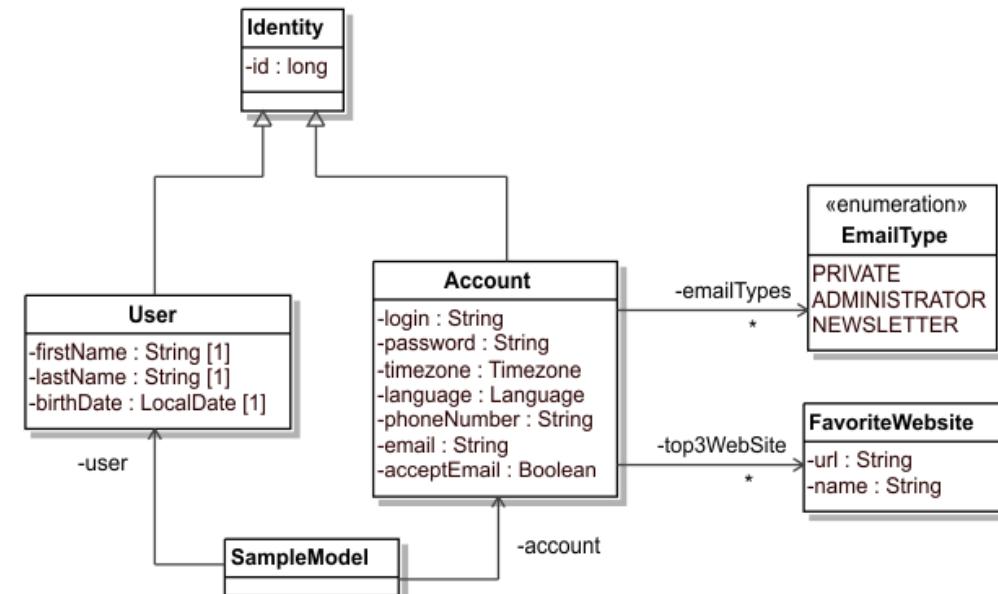
Timezone GMT+00:00

Submit

My Account

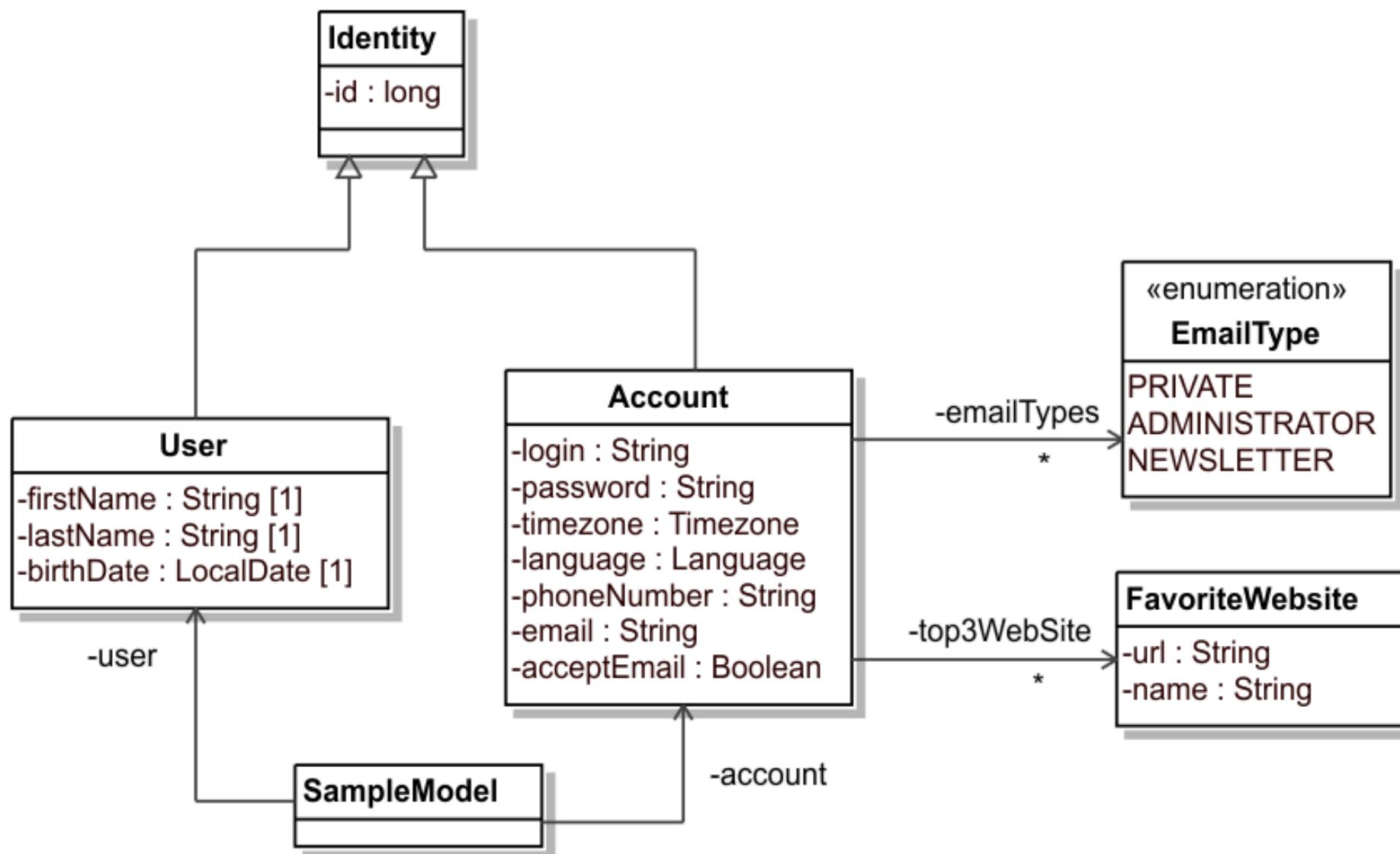


«enumeration»	
SampleFieldId	
USER_ID	
FIRST_NAME	
LAST_NAME	
FULLNAME	
BIRTHDATE	
ACCOUNT_ID	
EMAIL	
LANGUAGE	
TIMEZONE	
PHONE_NUMBER	
EMAIL_ACCEPTED	
EMAILS_PREFERENCES	
FAVORITE_SITE_NAME_1	
FAVORITE_SITE_NAME_2	
FAVORITE_SITE_NAME_3	
FAVORITE_SITE_URL_1	
FAVORITE_SITE_URL_2	
FAVORITE_SITE_URL_3	
LOGIN	
PASSWORD	

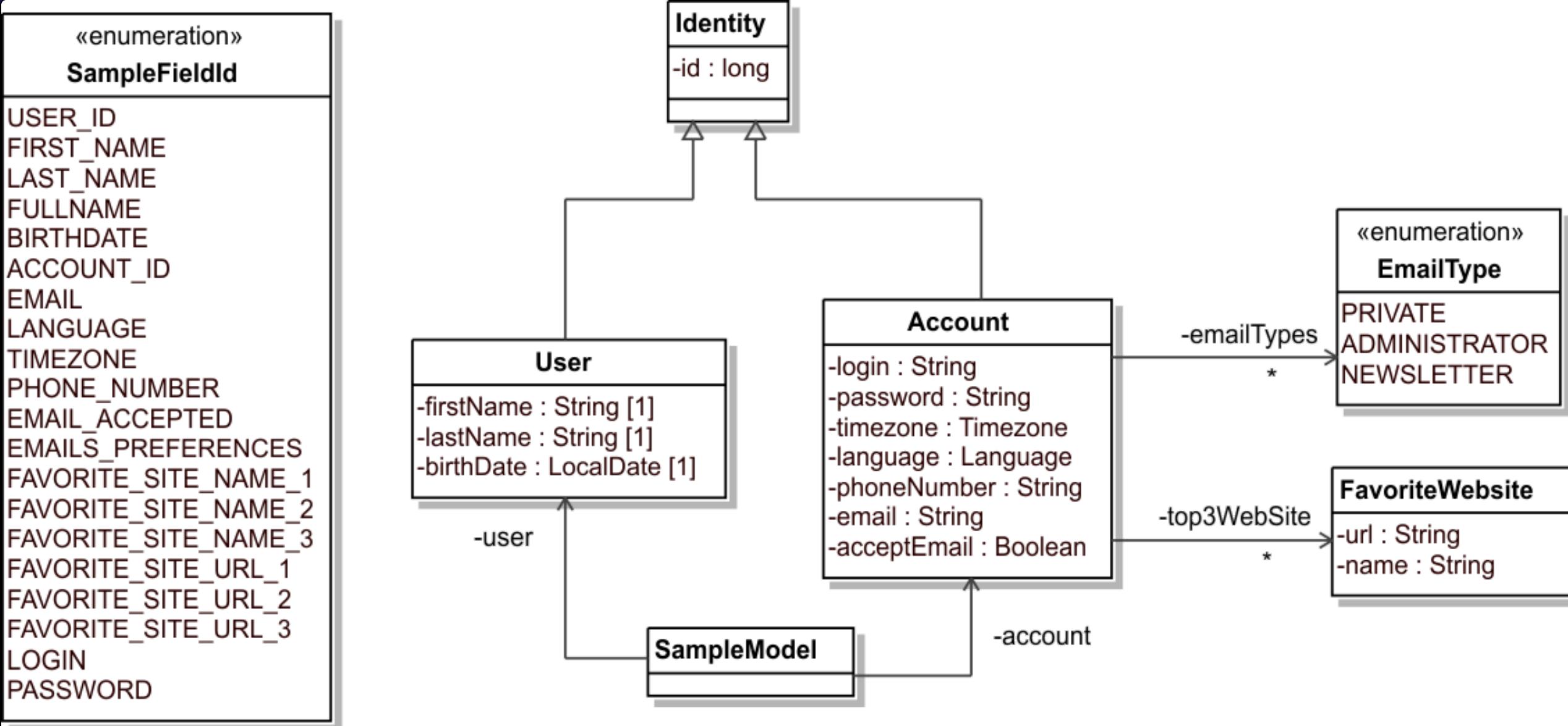


A BIT OF DECORATION

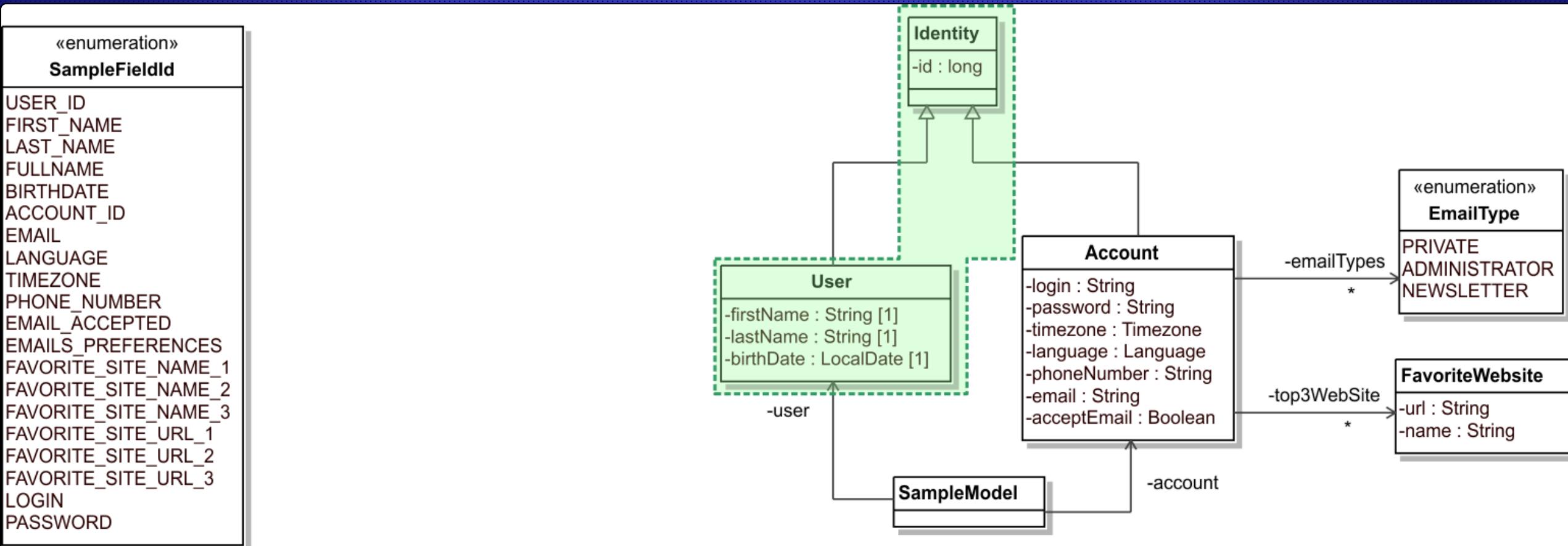
LET'S START WITH THIS MODEL...



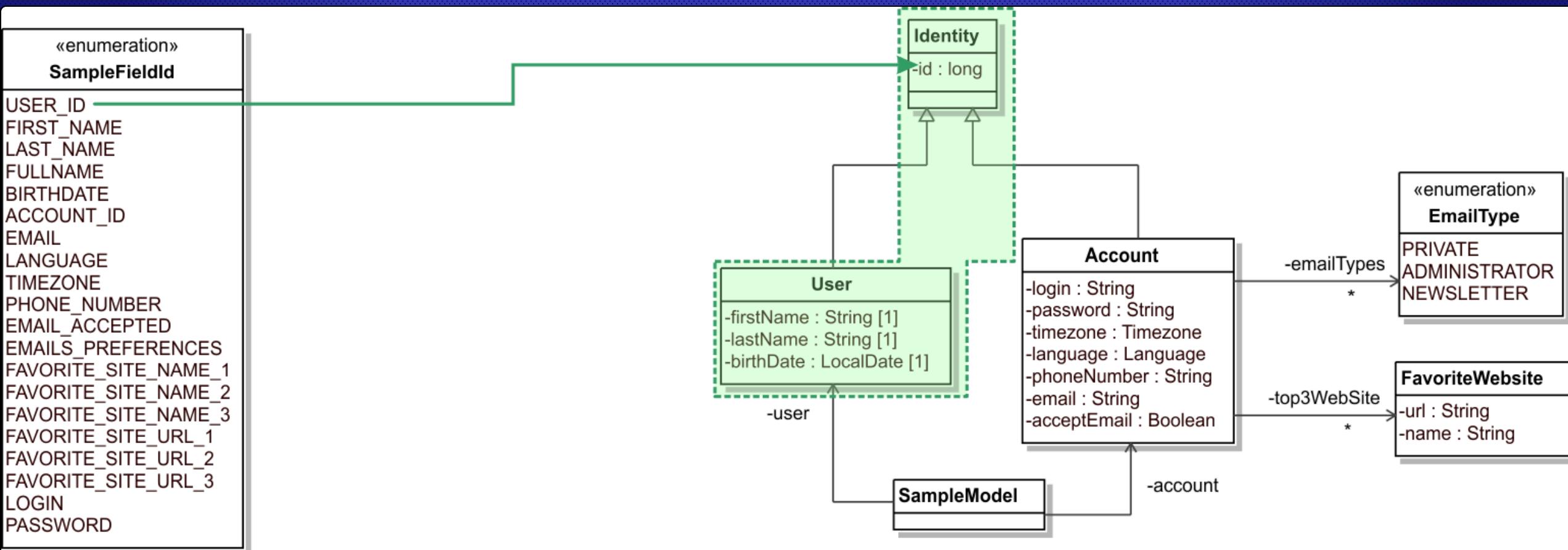
... AND TRANSFORM IT TO A KEY-VALUE MODEL



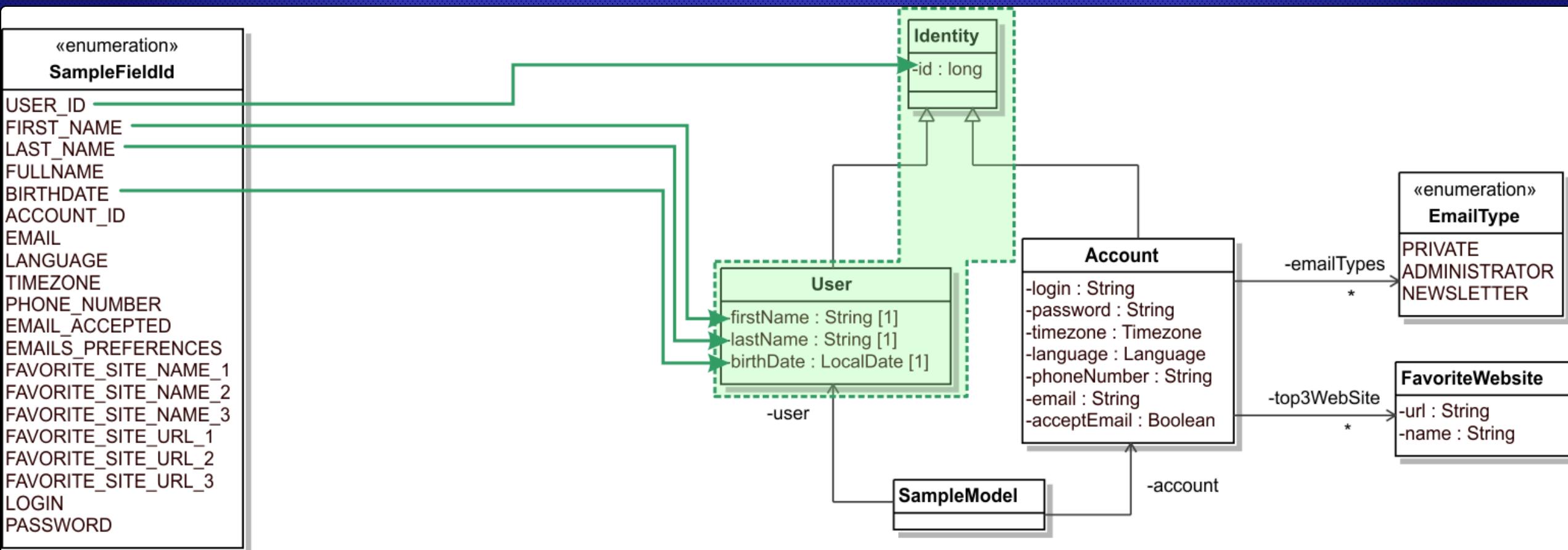
USER MODEL



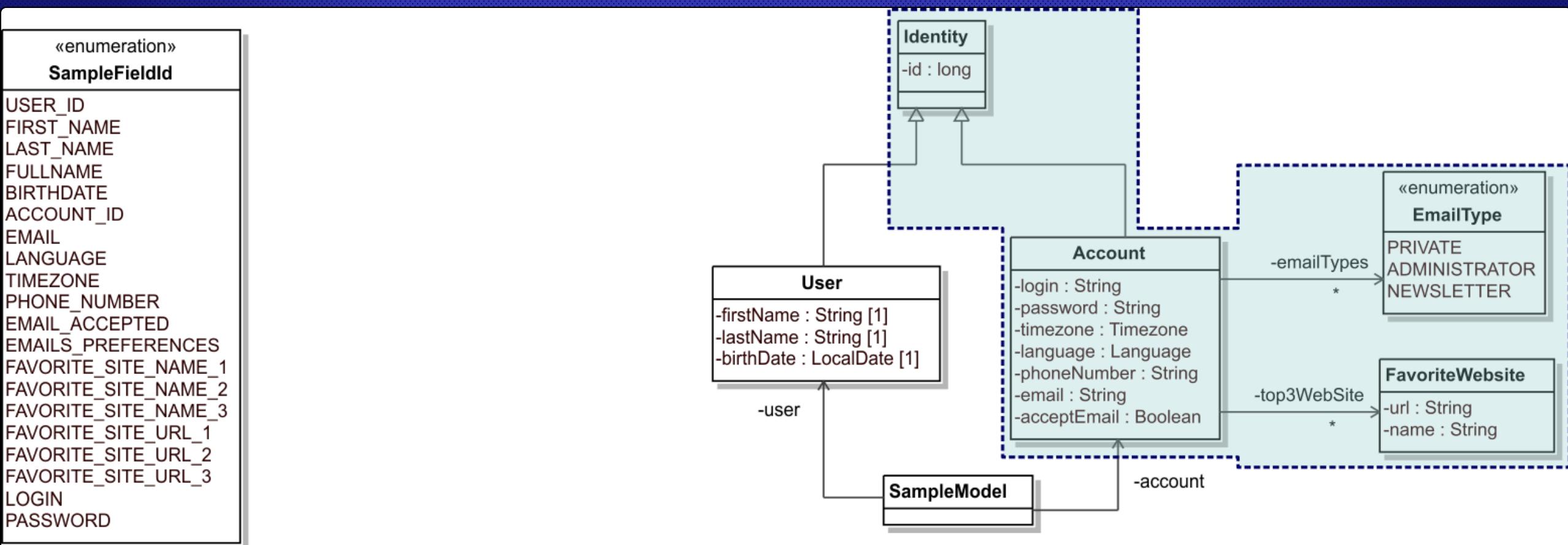
USER MODEL



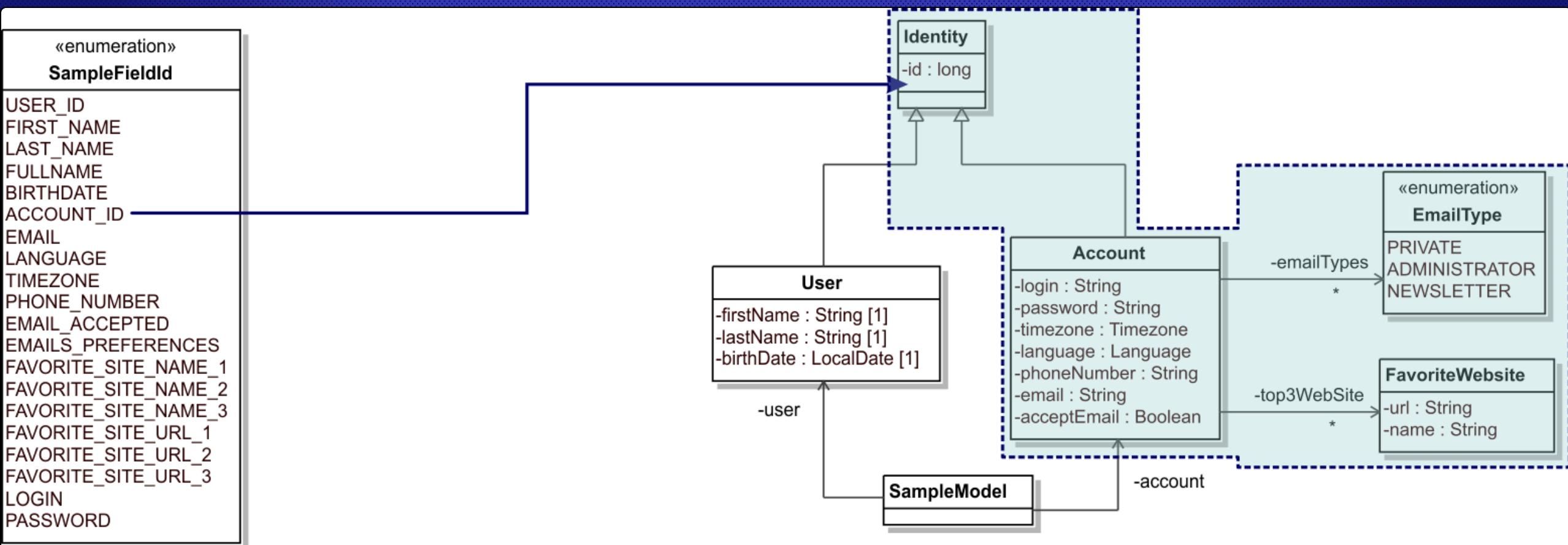
USER MODEL



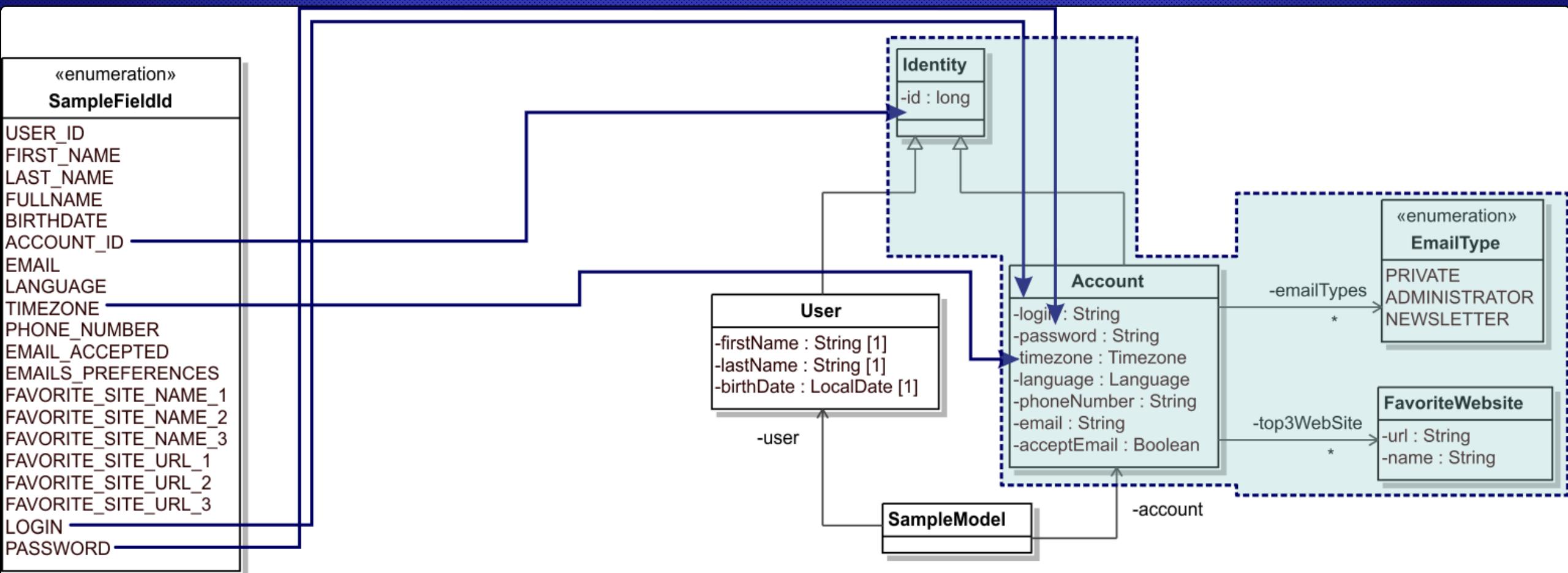
ACCOUNT MODEL



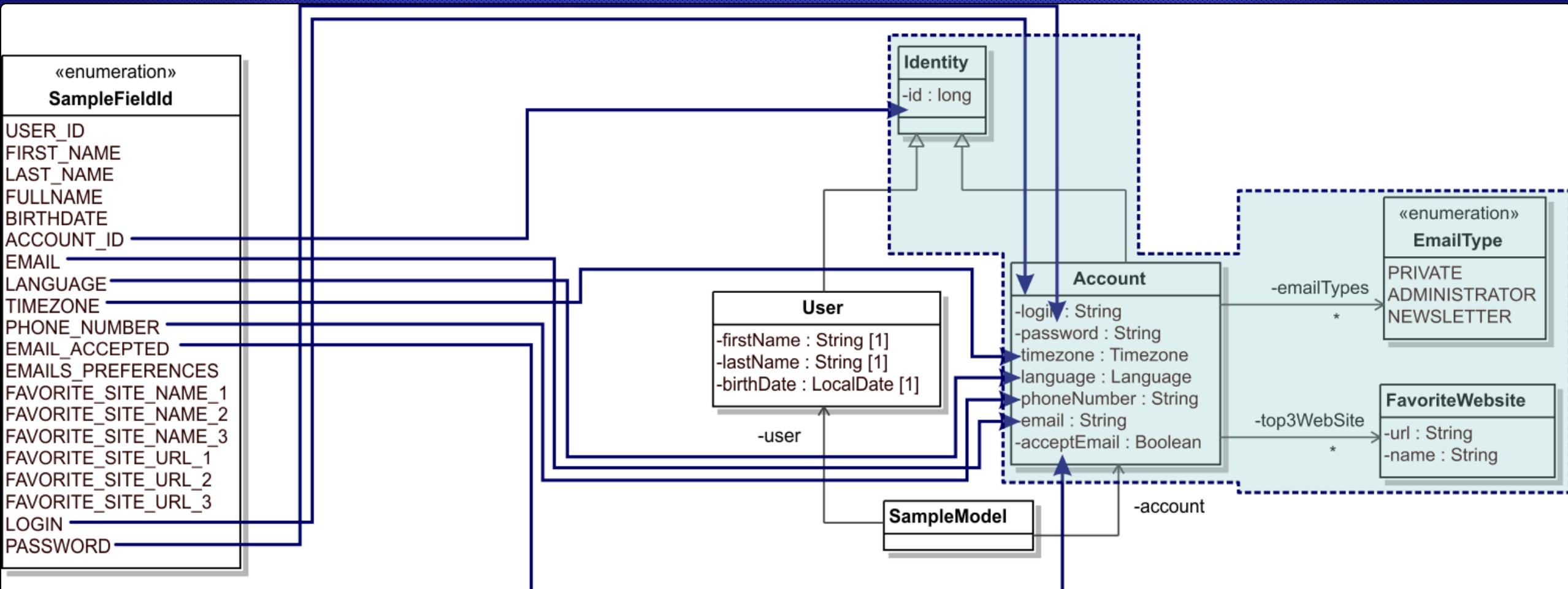
ACCOUNT MODEL



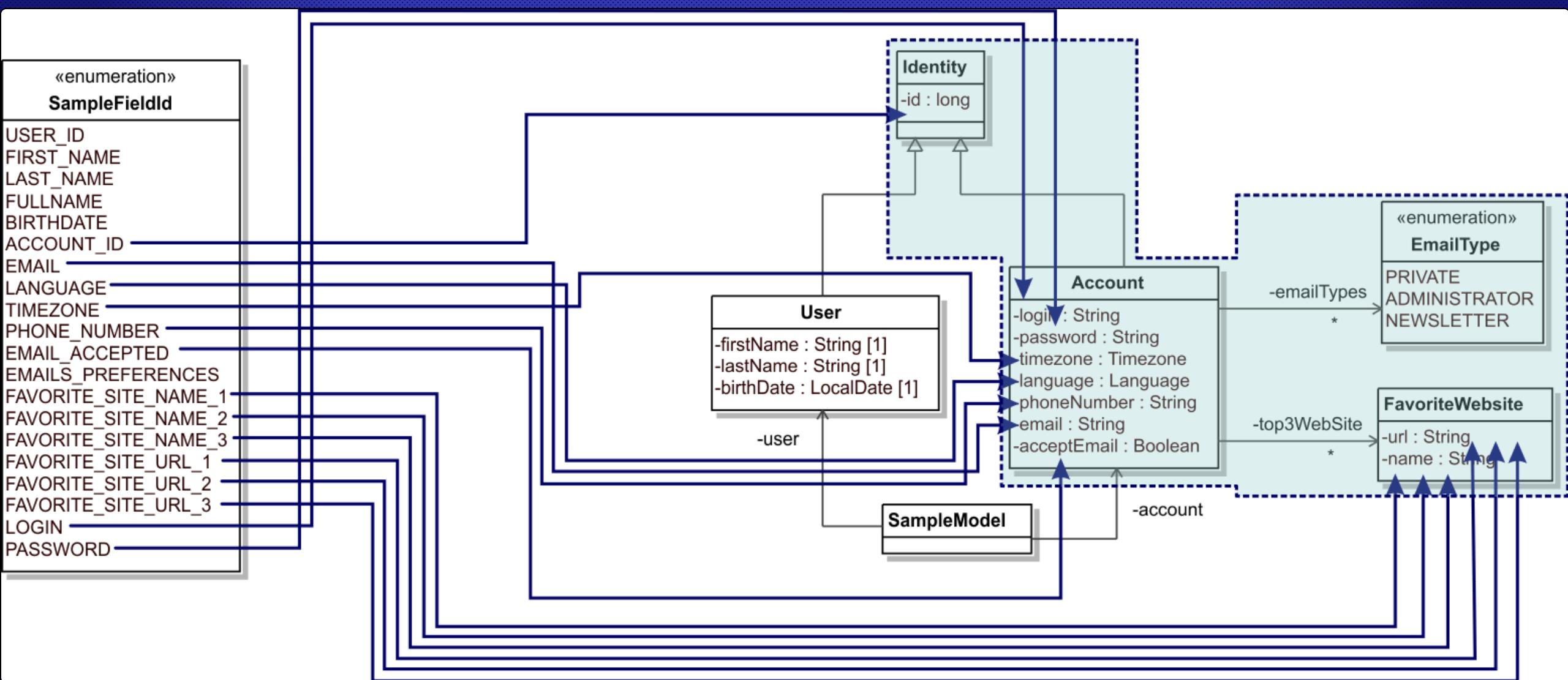
ACCOUNT MODEL



ACCOUNT MODEL



ACCOUNT MODEL



FROM DOMAIN MODEL TO KEY-VALUE MODEL

- Step 1 : create one Enumeration with all keys
- Step 2 : annotate your domain model
- Step 3 : generate the key value model

LET'S SEE THE DOMAIN MODEL & THE KEY-VALUE MODEL

STEP 1 : ONE ENUMERATION WITH ALL KEYS

- one literal for each key
- position (optional) : used when collections are denormalized
- tag (optional) : used to add meta-information

```
public interface FieldId {  
  
    int position();  
  
    List<TagId> tags();  
  
}
```

STEP 2 : ONE ANNOTATION TO ANNOTATE THE DOMAIN

- one method to retrieve the FieldId
- one method to retrieve the Constraint

```
@Path  
@Repeatable(SamplePaths.class)  
@Retention(RetentionPolicy.RUNTIME)  
public @interface SamplePath {  
  
    SampleFieldId field();  
  
    SampleConstraint constraint() default NONE;  
}
```

STEP 3 : ANNOTATE YOUR DOMAIN MODEL

- attribute, getter or setter can be annotated
- getter and setter must follow Java naming conventions

```
@SamplePath(field = SampleFieldId.FIRST_NAME)
private String firstName;

@SamplePath(field = SampleFieldId.LAST_NAME)
private String lastName;

@SamplePath(field = SampleFieldId.BIRTHDATE)
private LocalDate birthDate;

public User() {
}

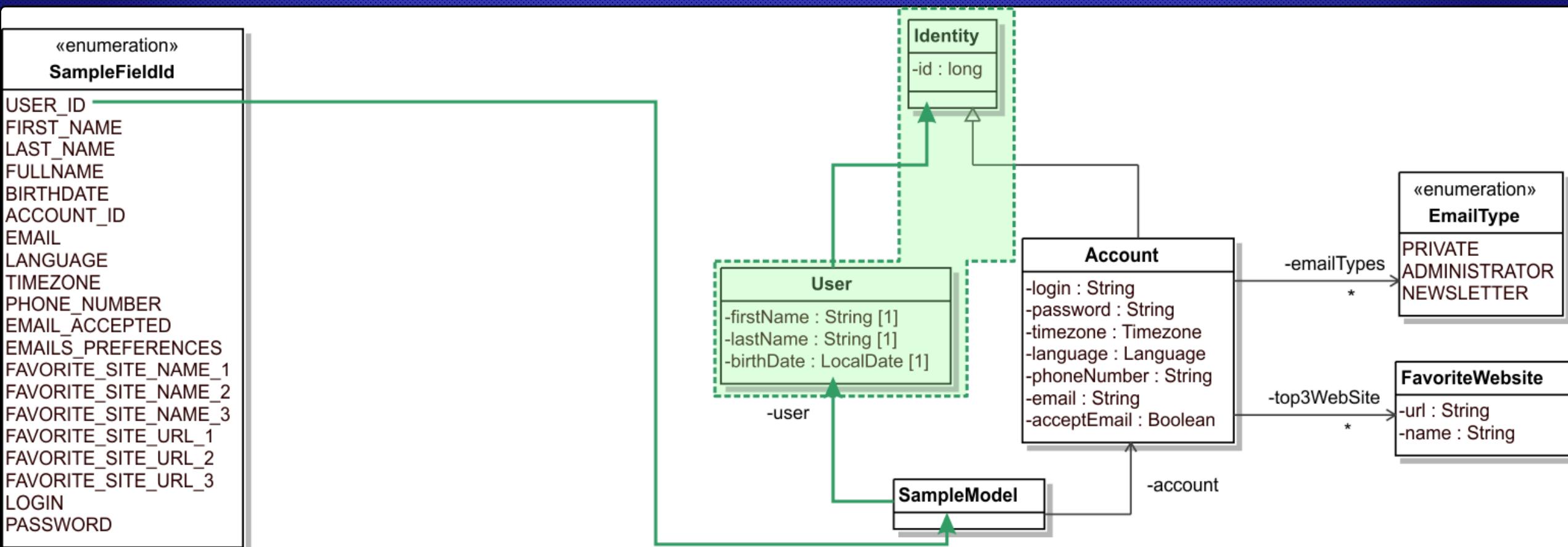
@SamplePath(field = SampleFieldId.FULLNAME)
public String getFullName() {
    return firstName != null && lastName != null ? firstName + " " + lastName : null;
}
```

STEP 4 : GENERATE THE KEY VALUE MODEL

```
interface FieldModel extends Iterable<Map.Entry<FieldId, Object>> {  
  
    <T> T get(FieldId fieldId);  
  
    <T> void set(FieldId fieldId, T value);  
  
}  
  
@Generated(value = "org.modelmap.gen.ModelMapGenMojo", comments="generated from org.modelmap.sample.model.SampleModel")  
class SampleModelWrapper implements FieldModel {  
  
    private final SampleModel model;  
  
    public SampleModelWrapper(SampleModel model) { this.model = model; }  
  
    @Override  
    public <T> T get(FieldId fieldId) {  
        // key-value mapping code here  
        return null;  
    }  
  
    @Override  
    public <T> void set(FieldId fieldId, T value) {  
        // key-value mapping code here  
    }  
}
```

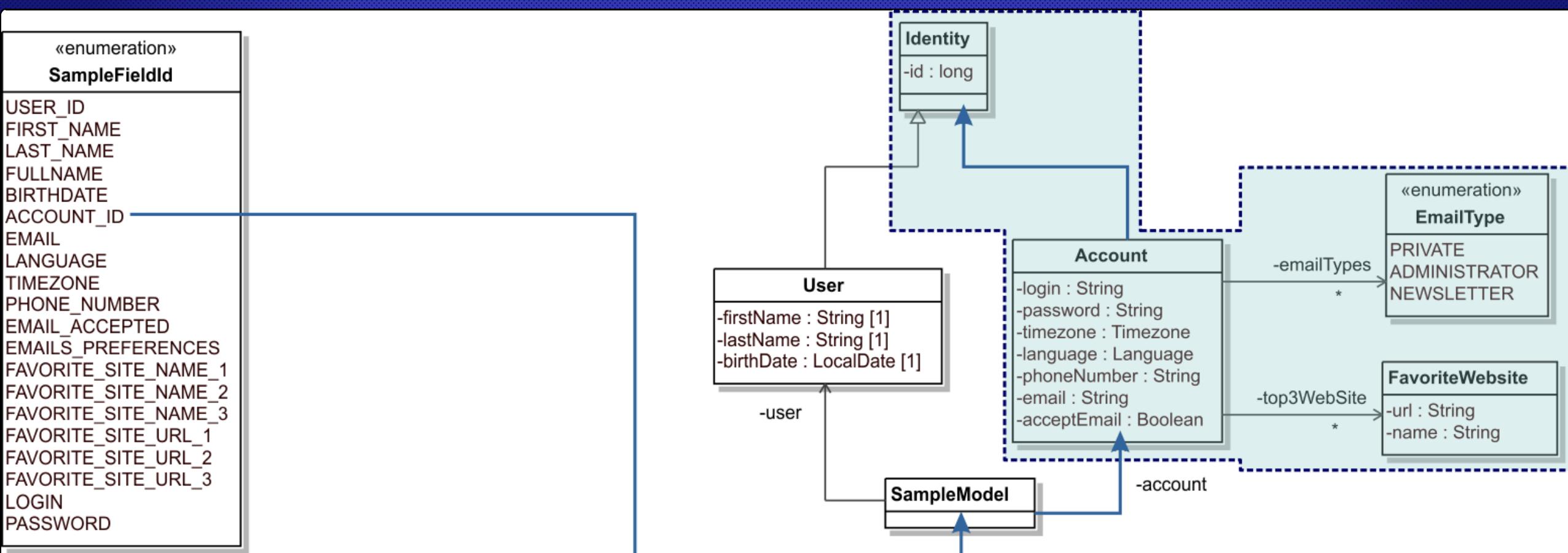
... JUST ONE LITTLE ISSUE.

ACCOUNT & USER ID



sampleModel.getUser().getId()

ACCOUNT & USER ID



sampleModel.getAccount().getId()

MANY KEY FOR A SINGLE ATTRIBUTE

- Each key must have a unique path to retrieve its value
- How to retrieve the right value for each field?
- getAccount().getId()? getUser().getId()?

```
public enum SampleFieldId implements FieldId {  
    USER_ID(USER),  
    ACCOUNT_ID(ACCOUNT),
```

SOLUTION : CONSTRAINTS

One for each field, specify which getter to use

```
import org.modelmap.core.PathConstraint;

public enum SampleConstraint implements PathConstraint {
    NONE(""), //
    USER("getUser()"), //
    ACCOUNT("getAccount()"), //
};
```

ADD A CONSTRAINT FOR EACH ANNOTATION

- the code generator only use the path with the specified getter
- each field is now mapped correctly

```
public class Identity {  
  
    @SamplePath(field = SampleFieldId.ACCOUNT_ID, constraint = SampleConstraint.ACCEPTED)  
    @SamplePath(field = SampleFieldId.USER_ID, constraint = SampleConstraint.USER)  
    private long id;  
  
    public void setId(long id) {  
        this.id = id;  
    }  
  
    public long getId() {  
        return id;  
    }  
}
```

LET THE MAGIC BEGIN ...

CODE GENERATOR UNDER THE HOOD

- Execution is done using a maven plugin
- Annotated properties are processed using Java Reflection
- Generate handwritten style code, easy to read
- Enable JIT optimisations by combining lambdas and enumerations
- Lightweight pattern, designed for performance

GENERATION OUTPUT (1/3)

```
public interface FieldModel extends Iterable<Map.Entry<FieldId, Object>> {  
  
    <T> T get(FieldId fieldId);  
  
    <T> void set(FieldId fieldId, T value);  
  
    Stream<Map.Entry<FieldId, Object>> stream();  
  
    Stream<Map.Entry<FieldId, Object>> parallelStream();  
  
    void setAll(FieldModel source);  
  
    void clear();  
  
    void clear(TagId tag);  
  
}
```

GENERATION OUTPUT (2/3)

An Enumeration containing metadata : type, generics type

```
public interface FieldInfo {  
  
    FieldId id();  
  
    Class<?> type();  
  
    Class<?>[] genericTypes();  
  
}
```

GENERATION OUTPUT (3/3)

a CSV containing each field, path, and value type

	A	B	C
1	Path	Id	Type
2	samplemodel.getUser().getId()	USER_ID	long
3	samplemodel.getAccount().getId()	ACCOUNT_ID	long
4	samplemodel.getUser().getFirstName()	FIRST_NAME	String
5	samplemodel.getUser().getLastName()	LAST_NAME	String
6	samplemodel.getUser().getFullName()	FULLNAME	String
7	samplemodel.getUser().getBirthDate()	BIRTHDATE	LocalDate
8	samplemodel.getAccount().getEmail()	EMAIL	String

USAGES ARE UNLIMITED

YOU CAN USE JAVA 8 STREAM ON YOUR DOMAIN MODEL !

- FieldModel#stream() : a stream of key/value pair (FieldId/Object)
- FieldModel#parallelStream() : parallel stream and better performance
- Easy use a filter/map/reduce operations on a domain model

Could you write a domain model diff tool now ?

WHERE WE USE IT AT LESFURETS.COM

- Simple UI binding by replacing JSON model by a Dictionary (80% less code)
- Use JSON Dictionary for client/server RPC
- Batch and ETL with an email provider (90% less code)
- Persistence in cassandra (from scratch with 200 lines of code for a whole model)
- Injecting user data in selenium (40% less code)
- Mocking our domain model in thousand of unit tests

... everything in iterative fashion during the past 2 years.

FUTURE AND IDEAS

- Push as OSS our Cassandra serialisation code
- Push some Spark samples
- JEP 276: Dynamic Linking of Language-Defined Object Models

MODEL-MAP AVAILABLE ON GITHUB

- <http://github.com/lesfurets/model-map>
- Framework and examples
- Apache Licence
- Try it and contribute !

THANK YOU!



{{ softshake }} 2015
http://soft-shake.ch

@ GENEVE