PHARO
An environment for industry and research

Marcus Denker, Inria Evref

What is Pharo?

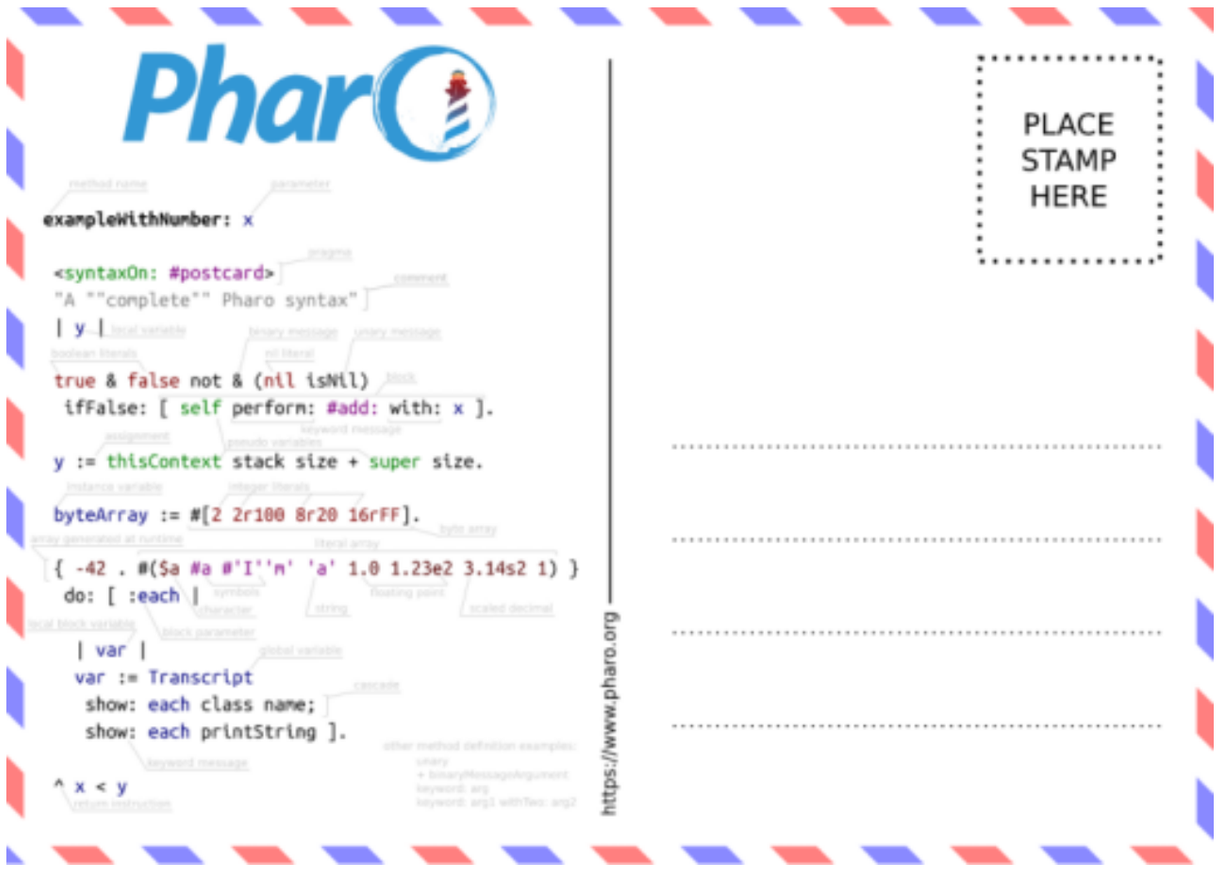
- Object Oriented
    - Classes
        - Single Inheritance
- Dynamically Typed
- Virtual machine, byte-code

So what is special?

- Sound not really exiting
- Ruby
- Python
- Original Java (but without types)

Essence of OO

- Radically Simple language
  - Syntax on a postcard



Reflective

- Written in itself
- Instead of text
- Classes are objects
- Methods are object
- Tools are objects and implemented in Pharo

Environment

- Object inspection
- Debugging
- Program and environment are one
  - of course in a controlled way

Features

- Advanced Runtime Reflection
- Pure object-oriented approach
    - Software as objects
    - First Class Variables
- Closures with non-local returns
- Fast resumable exceptions
- Traits

https://pharo.org/features

History: Smalltalk

- Smalltalk Project Xerox 1972-1978
- ST80
- Apple Smalltalk
- Squeak 1997-2008
- Pharo: started 2008

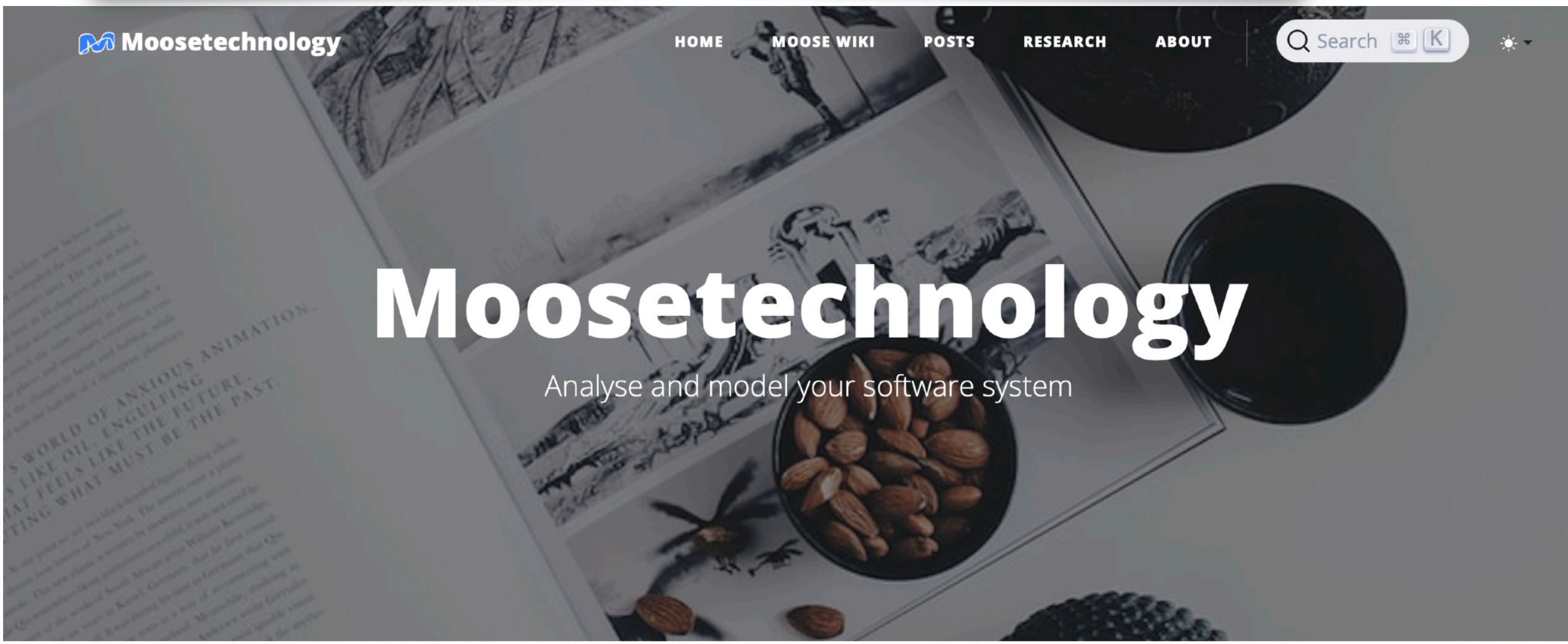
Not just a language

- “Pharo implements Smalltalk 80”
  - No!
- The goal of Pharo is to constantly improve the language and environment
  - While supporting users to migrate

Some Projects

- Both Research and Industrial

Sophisticated software analysis platform



Moose is a platform for software analysis.

It allows to re esent software system in a model, to query, manipulate, transform, visualize

r BSD/MIT.

https://modularmoose.org/

Cormas
Ressources naturelles et simulations multi-agents
Cirad • Web des SAVOIRS

https://cormas.cirad.fr/

Démarche | Logiciel | Applications | Bibliographie | Formation | Réseaux | Modèles

Bienvenue sur le site de Cormas

CORMAS = COmmon-pool Resources and Multi- Agent Simulations

Cormas est une plateforme de modélisation multi-agent (ABM) dédiée à la modélisation des relations entre les sociétés et leur environnement.

Cormas est destinée à faciliter la conception d'ABM, ainsi que le suivi et l'analyse des simulations. La plateforme est basée sur l'environnement de programmation VisualWorks qui permet de développer des modèles en Smalltalk.

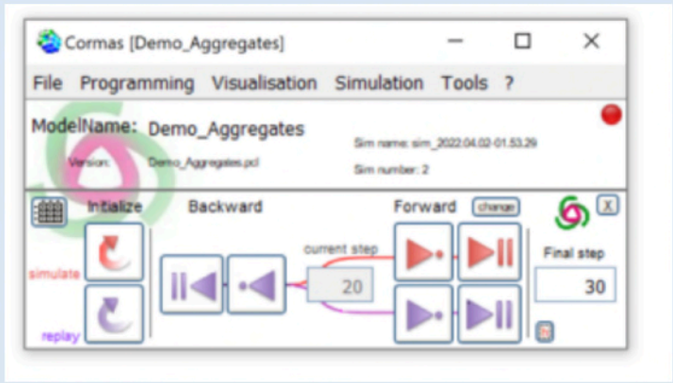
Cormas est un framework à partir duquel, par spécialisation et raffinement, les utilisateurs peuvent créer des entités spécifiques pour leur propre modèle.

cormas
COmmon pool Resources and Multi-Agent Simulations

Le logiciel Cormas, libre de droits (licence MIT), peut être téléchargé.

Découvrez la dernière version sortie en 2024.

Vous pouvez aussi consulter des exemples de modèles et des éléments de bibliographie

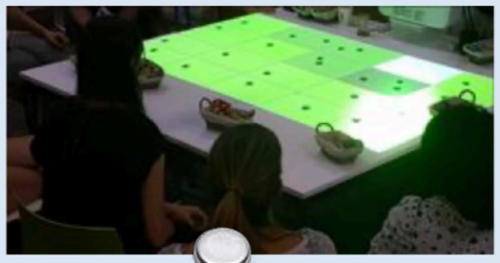


Interface principale de Cormas

- Pharo

Une nouvelle version de Cormas sous Pharo est en développement. Visiter le site

Nous proposons la démarche ComMod (Companion Modelling)



Session de RoHe… CN 2021 avec Cormas

Multi-agent simulation

ČVUT ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ ČVUT V PRAZE

CENTRE FOR CONCEPTUAL MODELLI AND IMPLEMENTATION

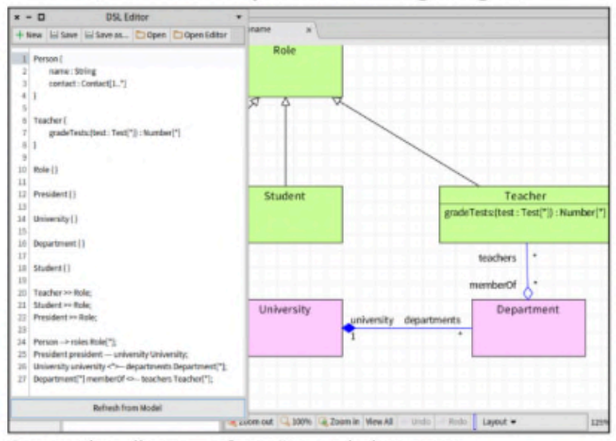
Pharo

OpenPonk

Modelling Platform

Conceptual modeling platform
for teaching, research, industry

DSL Support
(domain specific language)

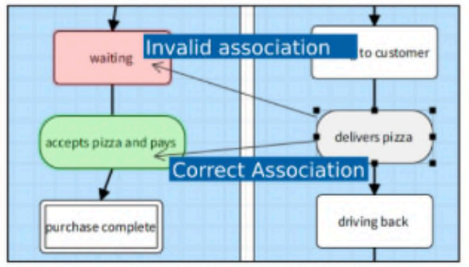


Generating diagrams for DSLs and vice versa
Some DSLs are part of OpenPonk,
but you can create your own.

OpenPonk goals

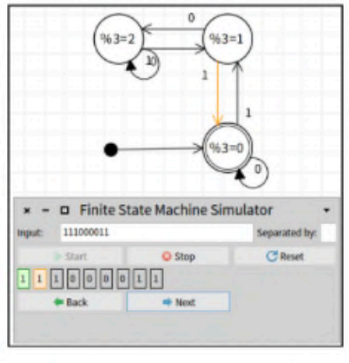
- Intuitive creation and interaction with models
- Support for existing and custom notations
- Extensibility for custom scenarios
- Open platform (MIT license)

Model validations



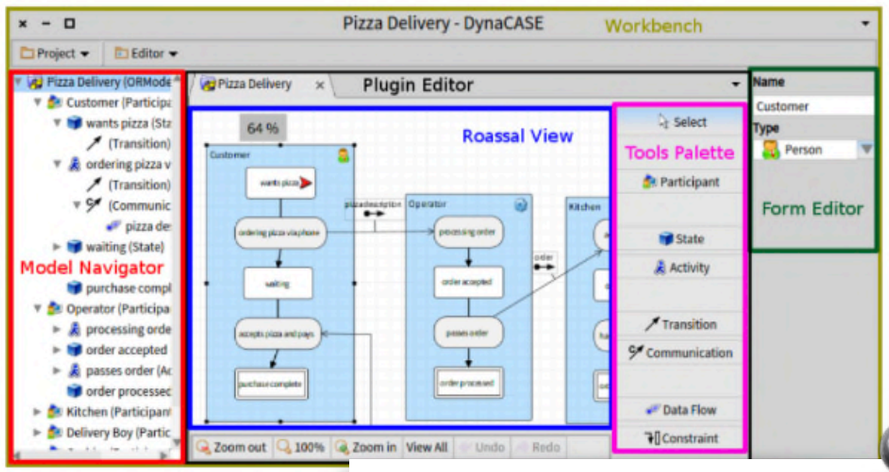
OpenPonk offers basic rule validation
during the creation process.
Complex validations are available for some notations.

Simulation and Visualization

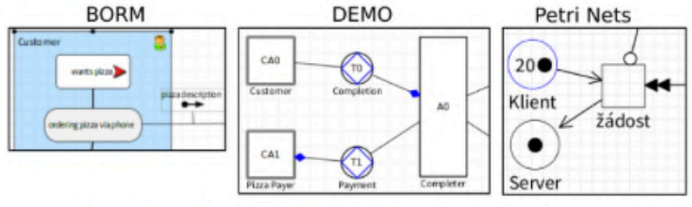


Step-by-step model simulation
and progress visualization.

Petri nets and State Automata
simulation is available.



Support for common and custom notations



UML class diagrams and
structural OntoUML

Finite State Automata

notation

openponk.org

If you are interested

t.cvut.cz

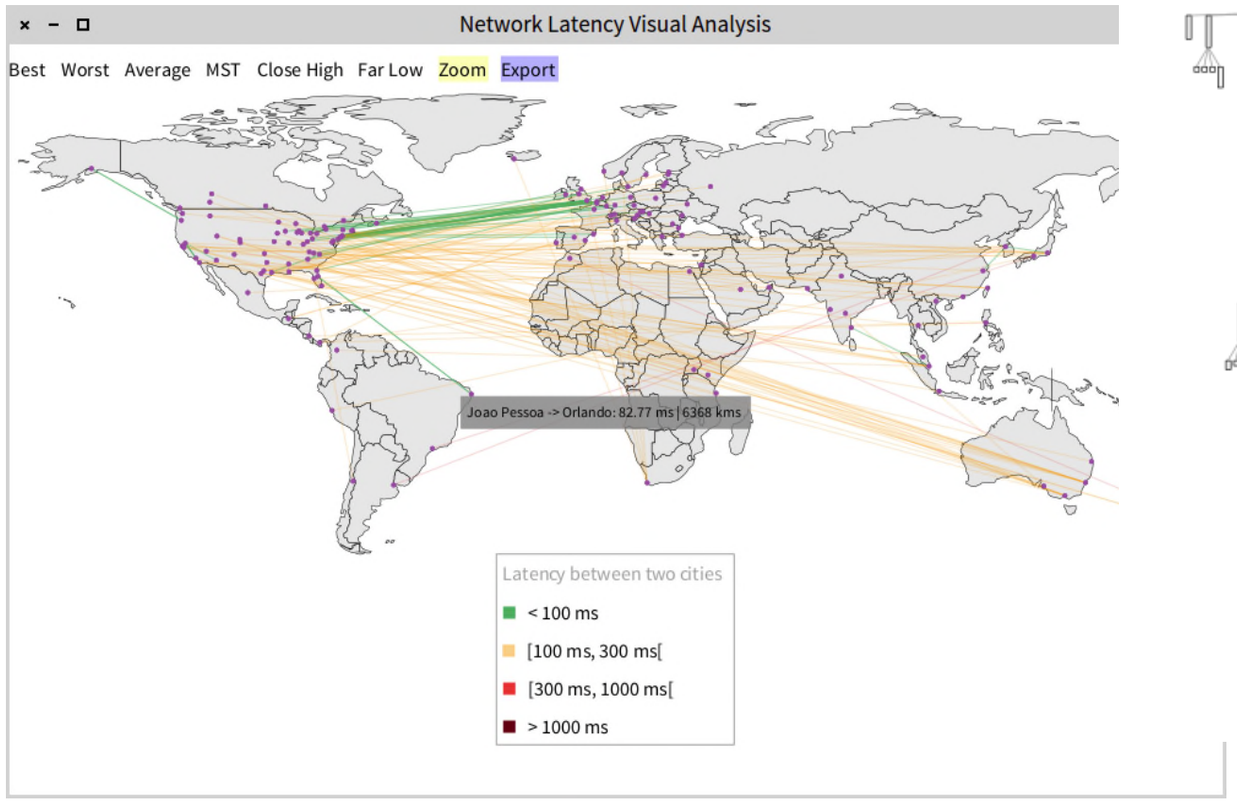
https://openponk.org/

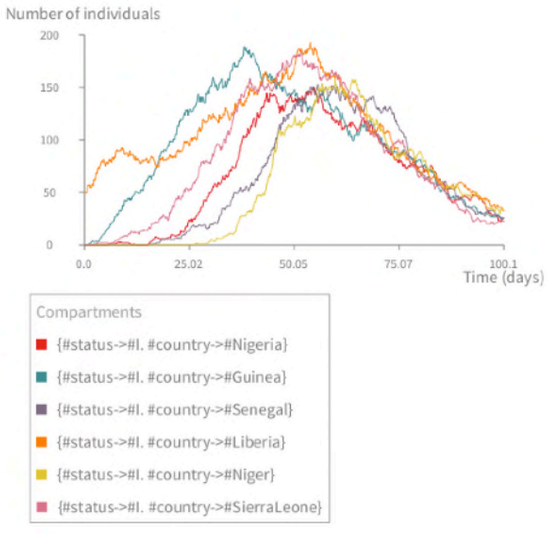
Roassal 3

- Amazing visual engine
- More than 10 years of development

- Some more links
  - https://www.youtube.com/watch?v=0h84-NZbfOg
  - https://www.youtube.com/watch?v=-Pk4q5oMdLo
  - https://www.youtube.com/watch?v=awPyBLWTTvk
  - https://www.youtube.com/watch?v=R2rLr7Z1b8Y
- Roassal 2 videos
  - https://www.youtube.com/watch?v=iXUZiFtnxK8
  -

Scripting visualization



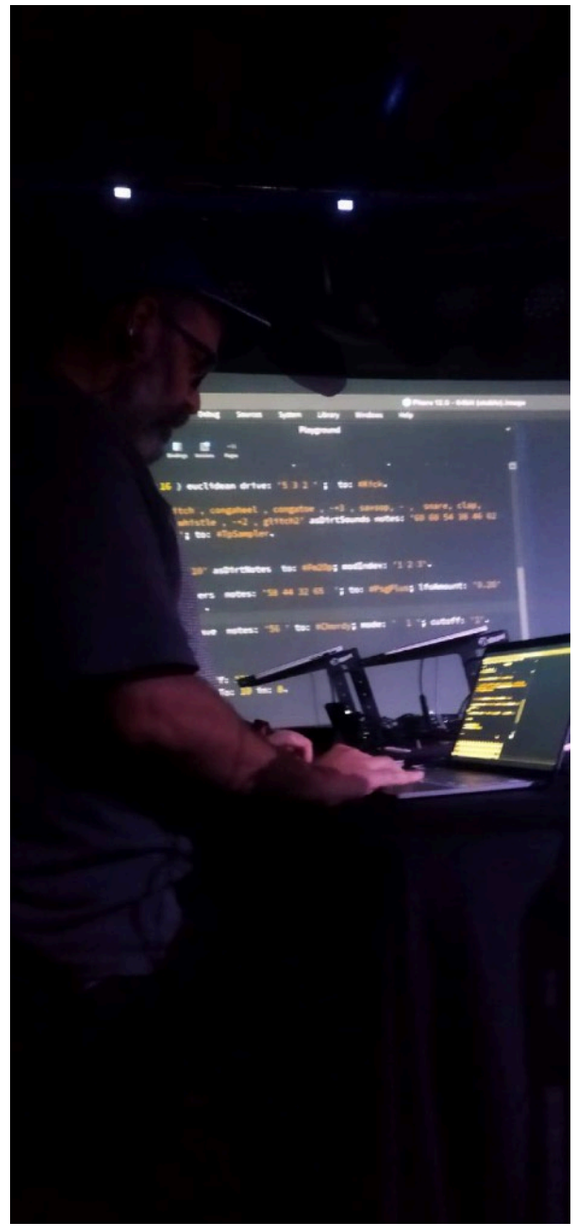


Live Music

D. Cipriano - DJ lucretio

- Live performance
- https://www.youtube.com/watch?v=4lhoYmI_ivs
- https://www.youtube.com/watch?v=S2Dff90aYDI
- Coypu and Phausto: Pharo live music
- https://github.com/lucretiomsp

ESUG 2024 https://www.youtube.com/watch?v=zP1GVVrydOs



PharoMusic.image

Pharo Browse Debug Sources System Library Windows Help

Playground

Do it Publish Bindings Versions Pages

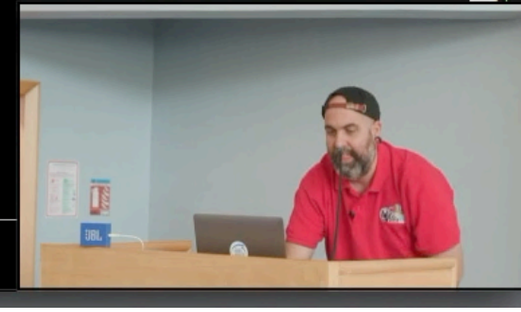
```
p := Performance uniqueInstance .
p performer: PerformerSuperDirt new .
p freq: 155 bpm.
p muteAll.
'sd' once.
'rekall' once.

'bd:3 ~ ~ ~ sd:2 ~ ~ ~ bd:3 d~ bd:4 ~ sd:2 ~ ~ rim ~'  forDirt to: #rhythm.

16 semiquavers sound: 'supergong' dirtNotes: #( 3 4 0) to: #synpop.
16 semiquavers sound: 'sd' dirtNotes: #(0 3 7 4 9) to: #snare.
4 breves sound: 'superhoover'; chords: 'f#-minor a-minor7 f#-minorsharp5 g-sus4'; to:
#hoover; gain: 0.89.
#(7 16) euclidean arpeggiate: 'f#-minor a-minor5 f#-major'; to: 'supersquare'.

" vertical fader to control the performance speed"
v6 := VerticalFaderFreq new.
v6 openInWindow.

```



Inspector on a Diction... Playground Playground Playground Playground


ICPianoKeyboard
Mixer
51
y Label
x Label
ICDarkKnob
0
0.000
100
ICLightKnob
0
0.000
100
ICDarkKnob2
0
0.000
100
ICDarkKnob3
0
0.000
100
ICDarkKnob4
0
0.000
100
ICDarkKnob5
0
0.000
100
ICDialKnob
0
0.000
100
ICSimpleKnob
0
0.000
100

Companies

Berger Levrault

Inria
INVENTEURS DU MONDE NUMÉRIQUE

Lifeware

ENSTA Bretagne

THALES

ZWEIDENKER

Yesplan

Lam RESEARCH

esug
EUROPEAN SMALLTALK USER GROUP

GemTalk SYSTEMS

SPESENFUCHS

projector software

feenk

netstyle.ch

inspired!

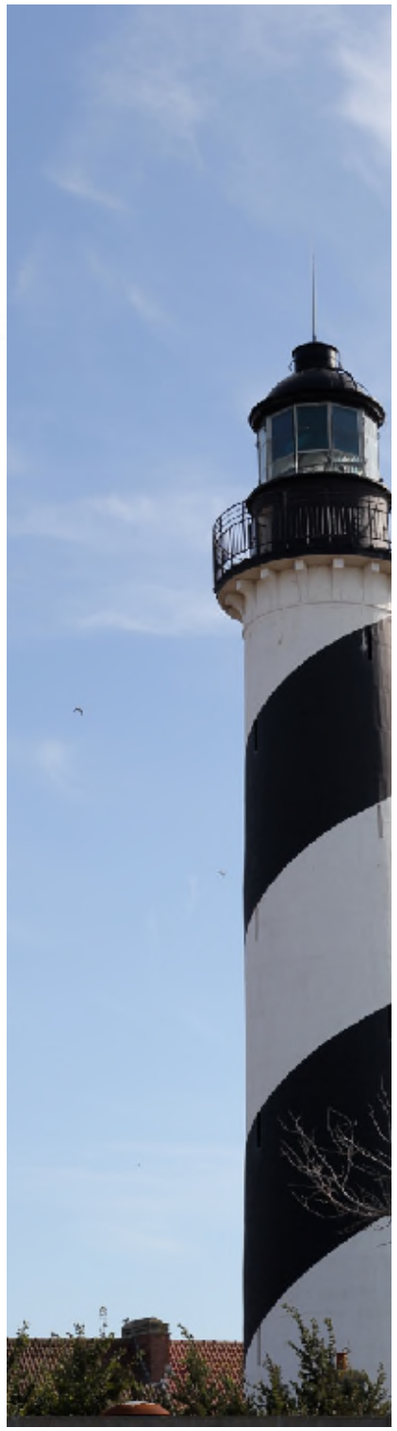
SRA GROUP

oco
oooco software company

telna

Université de Lille

FINWORKS
INFORMATION TECHNOLOGY SOLUTIONS



TELNA.

SOLUTIONS PRODUCTS PLATFORM RESOURCES COMPANY

CONTACT US

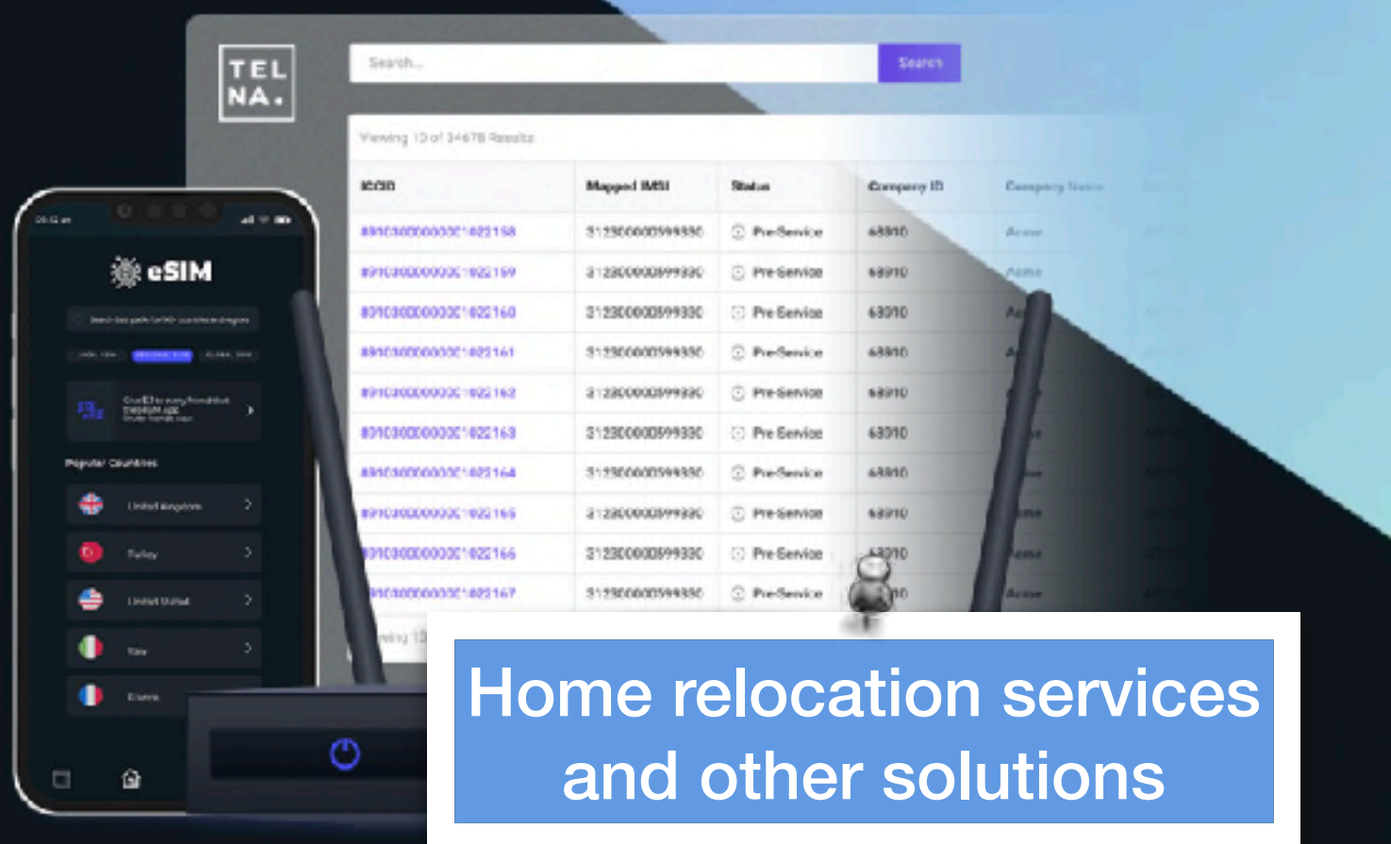
Global Cellular Connectivity platform built for platforms

Access endless possibilities through a single API with Telna, connecting you to cellular data coverage in over 200+ countries.

Designed for Connectivity Service Providers, our platform lets you prioritize your customers while we handle network infrastructure and relationships worldwide.

CONTACT SALES

aws EQUINIX KDDI Google Cloud



MobileTerminated ForwardSM

```
                    plan: SCCPGlobalTitle npISDN;
                    addr: destGT;
                    translation: 0;
                    yourself);
        yourself.

    forwardSM := ASN1Dictionary new
        at: 'sm-RP-DA' put: (ASN1Dictionary new
            at: 'imsi' put: destIMSI toTBCD;
            yourself);
        at: 'sm-RP-OA' put: (ASN1Dictionary new
            at: 'serviceCentreAddressOA' put: sourceSMSC toE164;
            yourself);
        at: 'sm-RP-UI' put: (ByteArray readHexFrom: '240c91948148136542000051107121955340 10c8329bfd0699e5ef3
        yourself.
    operationName := 'forwardSM'.

    dialogue := ZHLR endPoint  newDialogue.


    invoke := dialogue
                requestTcInvokeNamed: operationName parameters:  for
                class: 1 timeout: 500.
        invoke
        onReturn: [ ];
        onError: [];
        onReject: [];
        onReturnLast: [:result | ].

    dialogue
        requestTcBeginDestinationAddress: imsiAddr
        originatingAddress: mscAddr
        applicationContextName: '0.4.0.0.1.0.25.2' asAsn1Oid .
```

Do it d
Print it p

Extended search...
Do again j
Undo z

Free Software Mobile Communication solutions

Apply a display filter ... <⌘/>

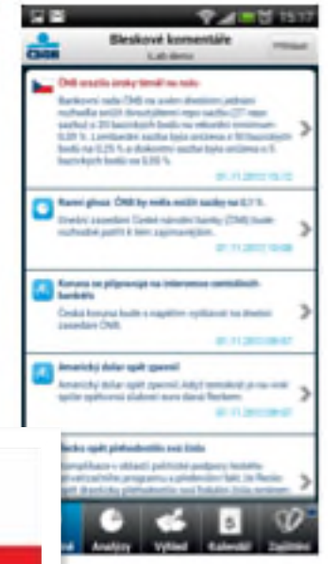
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	6	2000	GSM S..	236	invoke forwardSM

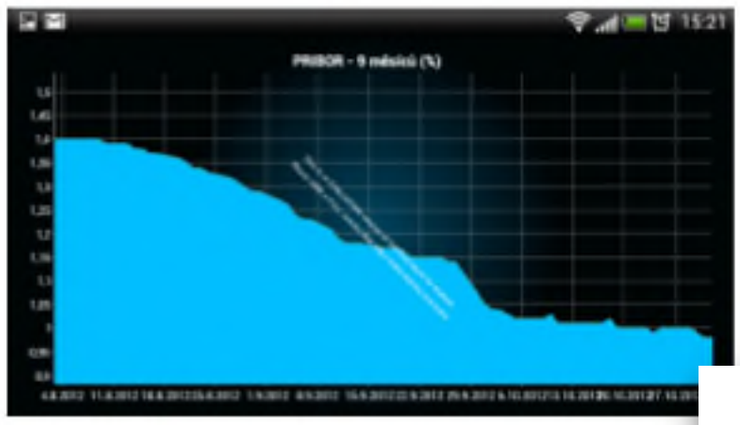
```
▸ Frame 1: 236 bytes on wire (1888 bits), 236 bytes captured (1888 bits)
▸ Linux cooked capture
▸ Internet Protocol Version 4, Src: 208.208.208.208, Dst: 208.208.208.208
▸ Stream Control Transmission Protocol, Src Port: 2905 (2905), Dst Port: 2905 (2905)
▸ MTP 3 User Adaptation Layer
▸ Signalling Connection Control Part
▸ Transaction Capabilities Application Part
▸ GSM Mobile Application
▾ GSM SMS TPDU (GSM 03.40) SMS-DELIVER
    0... .... = TP-RP: TP Reply Path parameter is not set in this SMS SUBMIT/DELIVER
    .0.. .... = TP-UDHI: The TP UD field contains only the short message
    ..1. .... = TP-SRI: A status report shall be returned to the SME
    .... 0... = TP-LP: The message has not been forwarded and is not a spawned message
    .... .1.. = TP-MMS: No more messages are waiting for the MS in this SC
    .... ..00 = TP-MTI: SMS-DELIVER (0)
  ▸ TP-Originating-Address - (4491884315624)
  ▸ TP-PID: 0
  ▸ TP-DCS: 0
  ▸ TP-Service-Centre-Time-Stamp
    TP-User-Data-Length: (16) depends on Data-Coding-Scheme
  ▾ TP-User-Data
      SMS text: Hello from Pharo
```

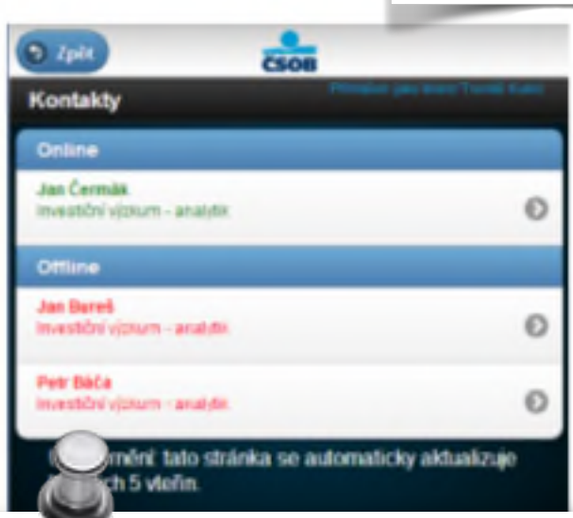
```
0000  62 69 48 04 01 00 00 09  6b 1e 28 1c 06 07 00 11   biH..... k.(.....
0008  86 05 01 01 01 a0 11 60  0f 80 02 07 80 a1 09 06   .......` ........
0010  07 04 00 00 01 00 19 02  6c 41 a1 3f 02 01 01 02   ........ lA.?....
0018  01 2c 30 37 80 08 62 02  43 68 10 91 50 f9 84 08   ..07..b. Ch..P...
0020  91 94 51 97 10 00 08 08  04 21 24 8c 91 94 81 48   ..Q..... .!$....H
0028  13 65 42 00 00 51 10 71  21 95 53 40 10 c8 32 9b   .eB..Q.q !.S@..2.
0030  fd 86 99 e5 ef 36 80 8a  0e cb df 80               .....6.. ....
```

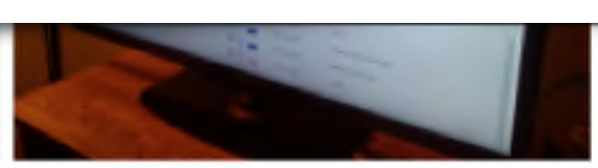
MobileTerminated ForwardSM

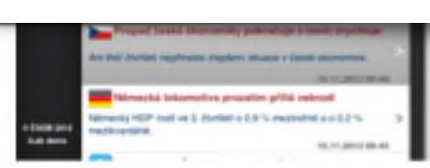
ČSOB







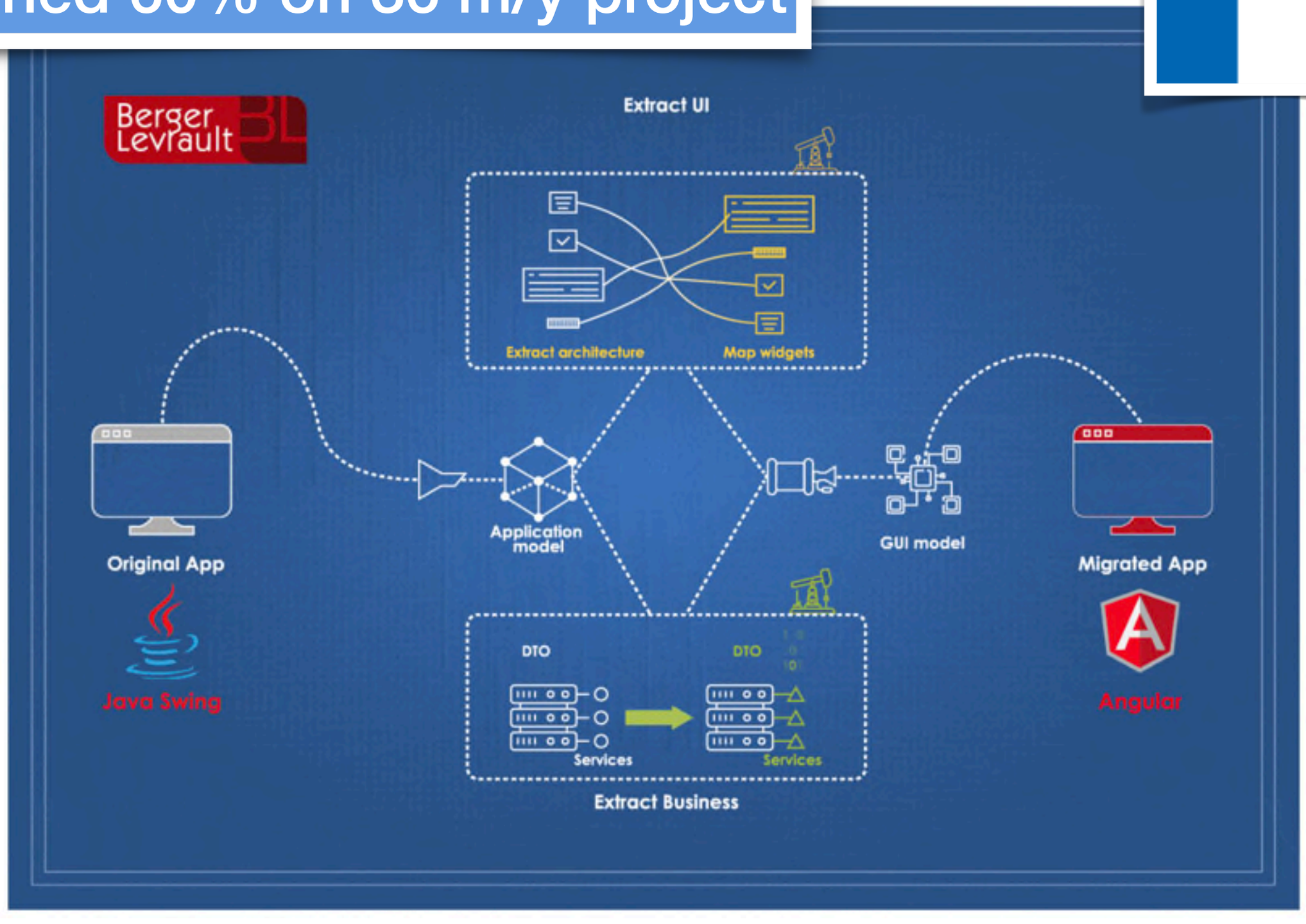




Developed in 3 months one dev
Sold to another bank
Java team estimated 2 year effort

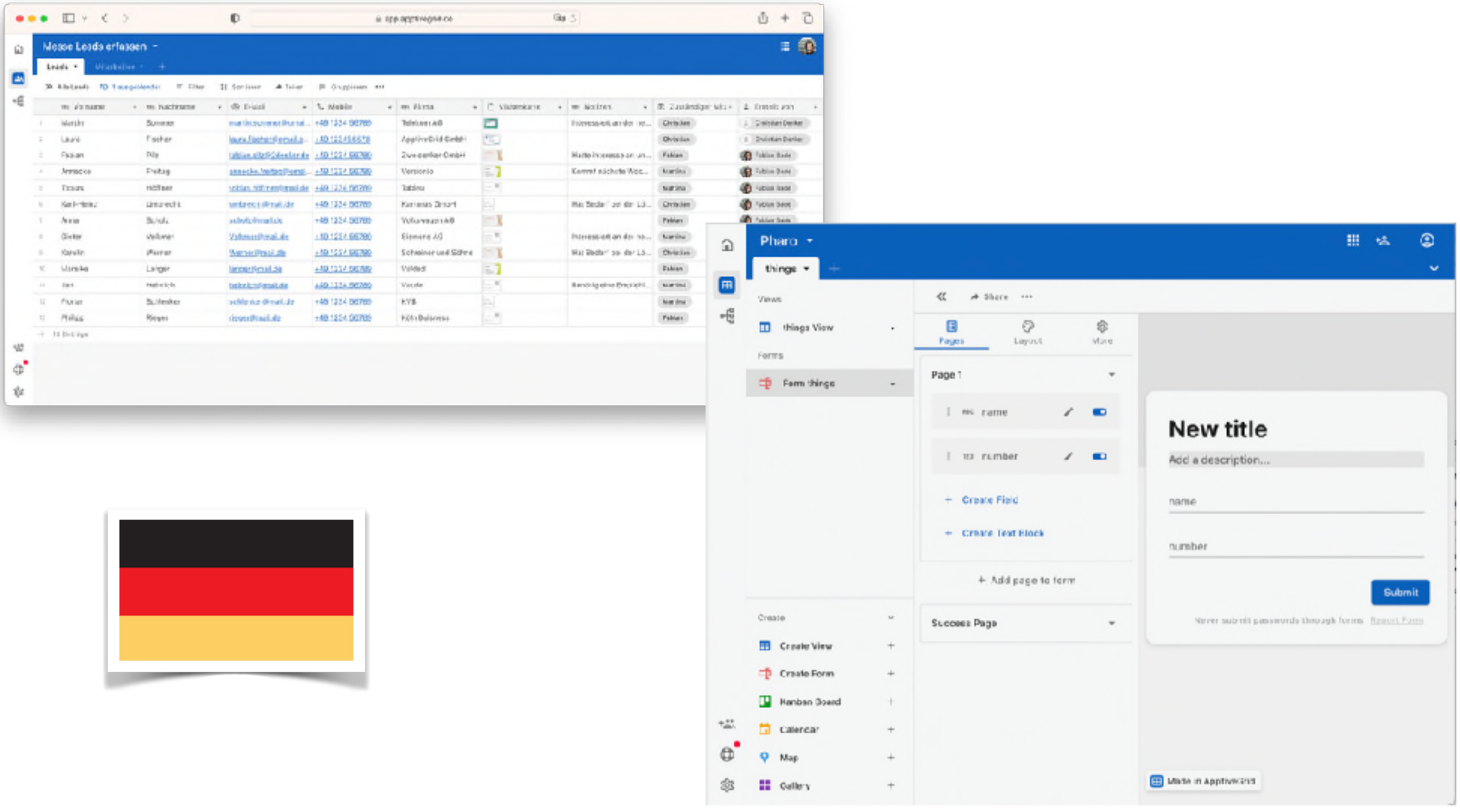
Application Migration for Berger-Levrault

Gained 60% on 36 m/y project



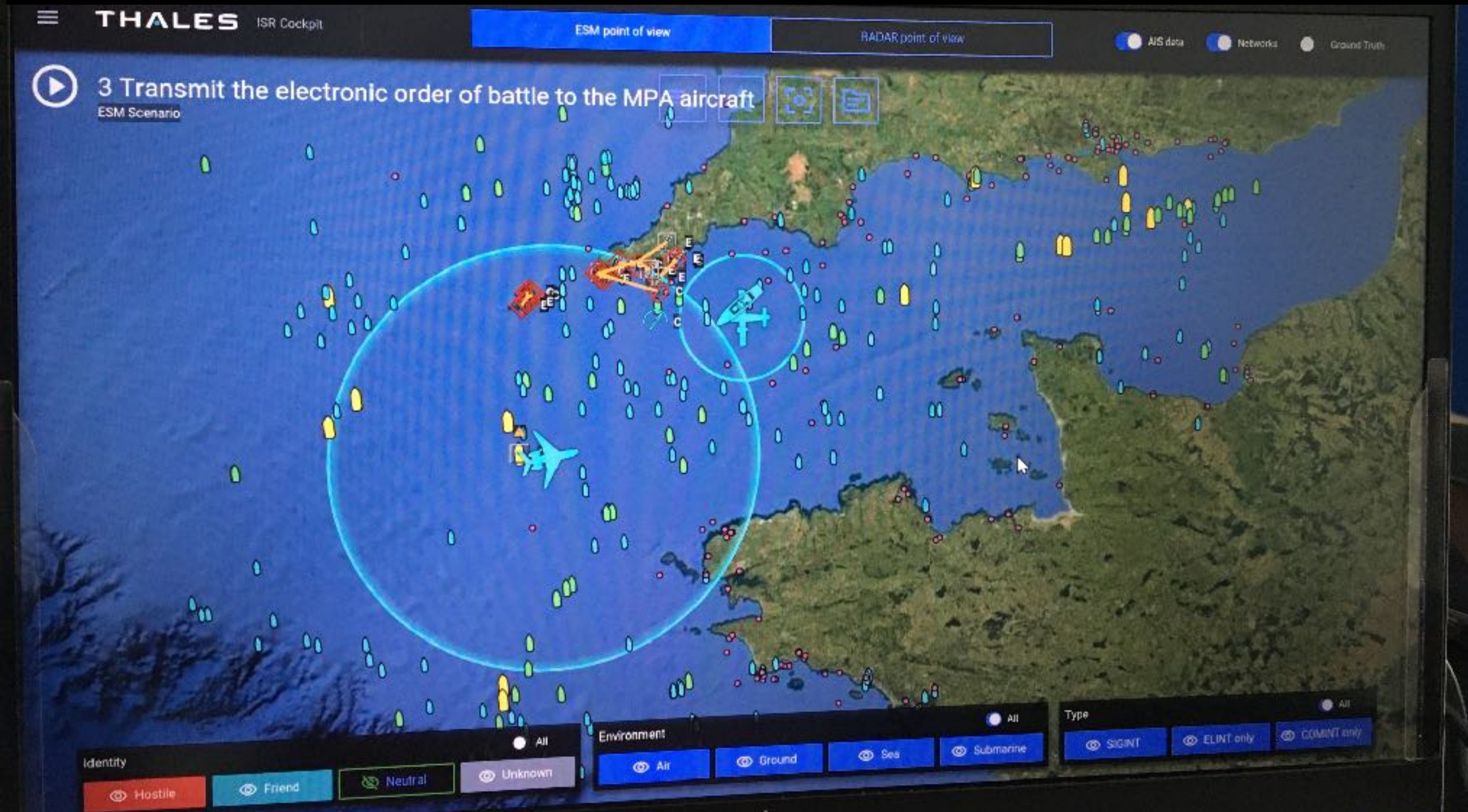
ApptiveGrid

Digitize and Automatize Business Processes



Sales support @ Thales

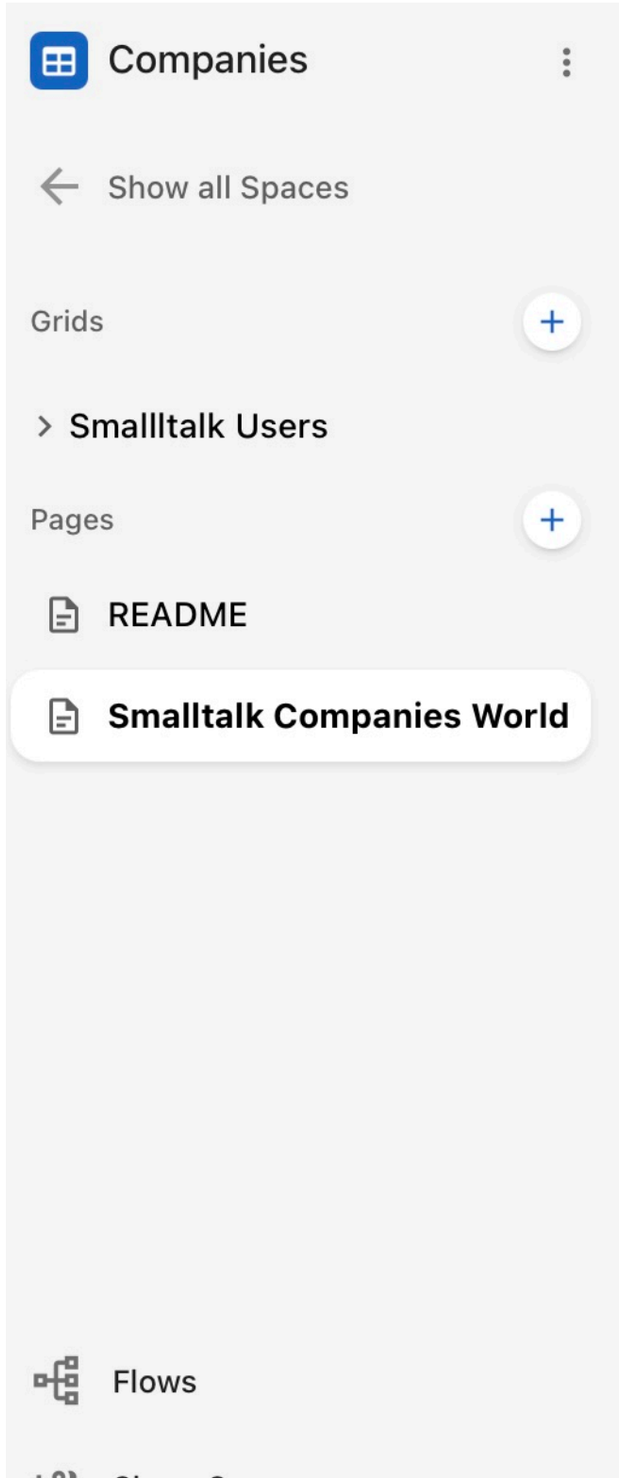
Geo-Based Maps



Feedback Loop

- Between Users and Pharo
- Both Research and Industry
- Not trivial !
    - Interesting Problem, both Engineering and Research

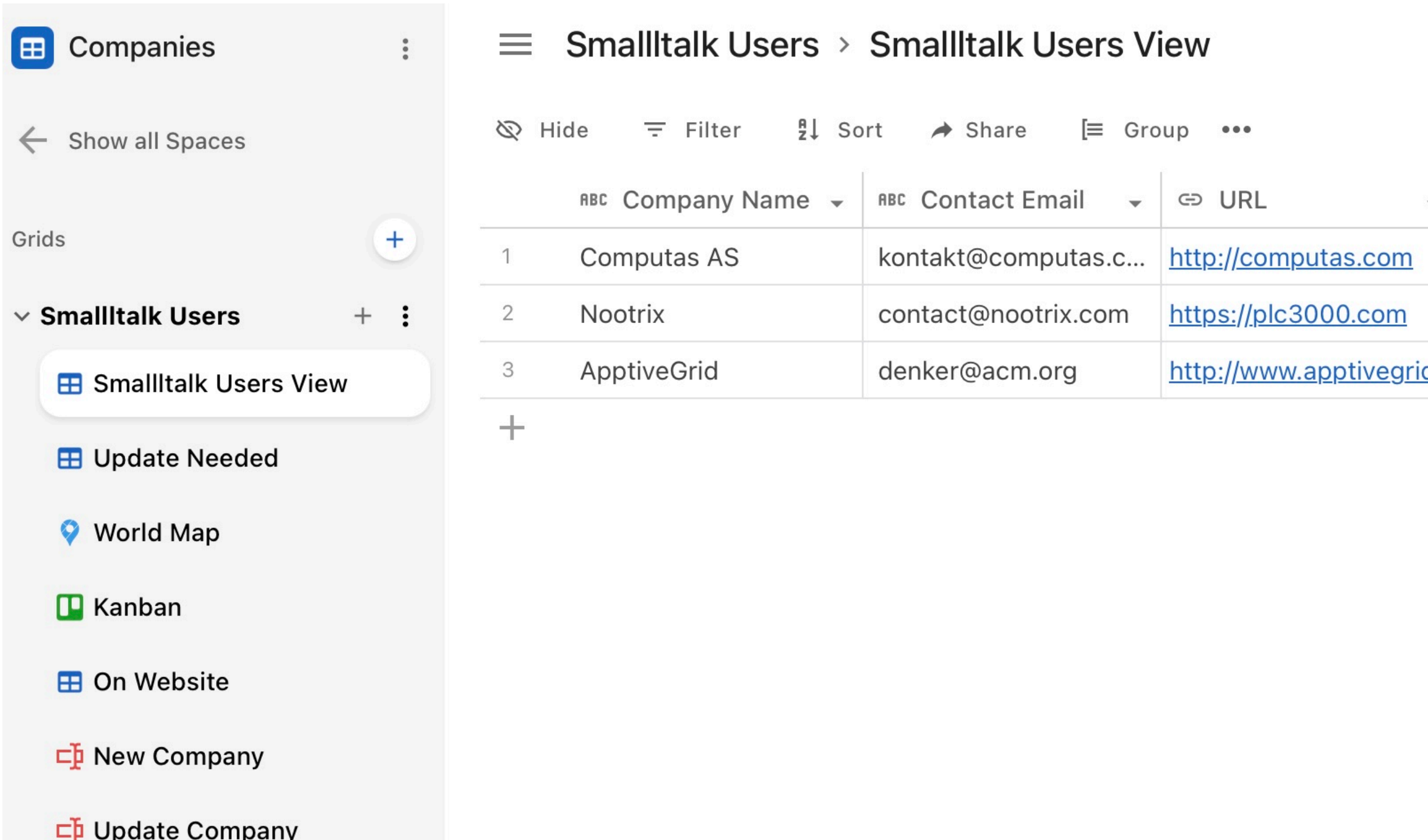
ApptiveGrid



ApptiveGrid

- Platform to automatise / solve problems for companies
- Data in Grids (Tables)
- Forms
- Views
- Workflows
- Web View

ApptiveGrid



What is Soil?

- Soil is an **Object-Oriented Database** implemented in Pharo
- ACID transactions, MVCC (append only + GC)
- Indexing: SkipList and B+Tree
- Goal: Simple yet powerful database making it easy to develop with, easy to debug with, easy to inspect, ...

Soil Properties

- No external database needed
    - Simplifies deployment
- MVCC (multi version concurrency control)
    - Data never changed on disk

Soil is Small (v3)

- 134 classes, ~1900 methods, ~8 k linesOfCode

```
Soil package definedClasses size.  "122"
Soil package methods size. "1686"
Soil package linesOfCode.  "6846"

SoilSerializer package definedClasses size.  "12"
SoilSerializer package methods size.   "301"
SoilSerializer package linesOfCode   "1276"
```

Soil and Pharo

- Example of Industrial <-> Pharo <-> Research Loop
- Was created to solve the real world problems of ApptiveGrid Data storage
- Result leads to interesting ideas when applying it to Pharo itself

Soil and Pharo

- Lots of Pharo IDE Problems are Database Problems
  - Code storage
    - source and compiled
  - Indexing for faster search
  - Code History

Pharo as a reflective Sytem

- Pharo is implemented in itself
- Has a model of itself
- Classes, Methods are objects

Current state

- Source is stored in .sources/.changes
- All queries just search
  - including full source search
- History: .changes , git, epicea

Sources/Changes

- Every compiled method has a pointer to .sources/.changes
- Stored in the last 5 bytecodes
- Every entry is an offset in the files

Need: Not in Image

- .sources is 40MB

- Would be a large chunk of the image

Need: History

- Having a per-accept history is nice
    - Does not depend on commits to git

Need: Transaction Log

- Pharo crashes and you did not save the image or stored the code in git

- All accepted methods / class comments / class definitions are in the .changes

- Replay is possible

Current problems

- Can not be shared by images, but can not be locked either
- Not thread safe
- Very simple caching
- **SLOW!**

Soil to store code

- Has transaction log
- Supports accessing old versions
- We could just store a Soil Object Pointer per CompiledMethod
- Thread safe

Soil for Indexing

- Soil has on-disk indexing. Can be useful:
  - Senders
  - Implementors
  - Full text search

Indexing

- SoilIndexedDictionary
  - SoilBTree
  - SoilSkipList
- key -> value
  - key has to be mappable to binary, sortable value (e.g. symbol). Fixed pre-defined size
  - value can be any object

Low Level: SoilIndex

- SoilIndex
  - binary key -> objectID
- Stored in 4Kb pages on Disk
- Data-page form a double linked list
  - fast iteration

Just using Soil

- The goal of Soil is to be a commercially usable OO Database for Pharo

- As we have seen: Just using it is very interesting

We can go further

- Here some ideas what we can do next
- Use Soil for even more
- Ideas for next Soil features
- Research ideas beyond

Store code

- Soil stores class description objects
  - no code of methods
- Storing code is an interesting experiment
  - useful e.g. when storing blocks (lambdas)
  - simplifies object Migration
  - simplifies evolution of Soil itself

Code history

- We do have code history of course, similar to how everyone has it
- .changes, Epicea, Git
    - Git is just dead text

- Reflective Model does not model history !

Can we do better?

- model the whole history of the code ?
- With indexing and fast queries
    - e.g find deprecation transforming methods
- store at AST granularity?

Transactions

- Transactional change as a language feature
- Reflective change is very brittle
- We already implement parts ad-hoc
  - e.g. Refactoring
  - Code loading

DB vs Language

- Every non-trivial program handles data
- Yet we, as programming language researchers, define it to be not our problem
- Is this really a good idea?

Pharo should be persistent and history aware