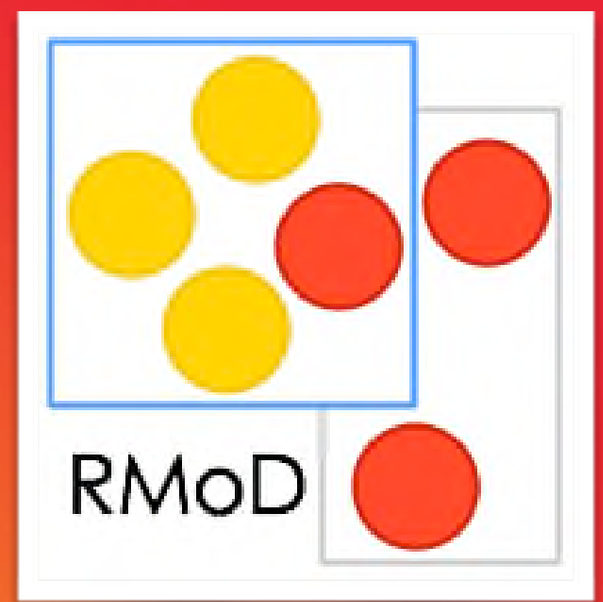
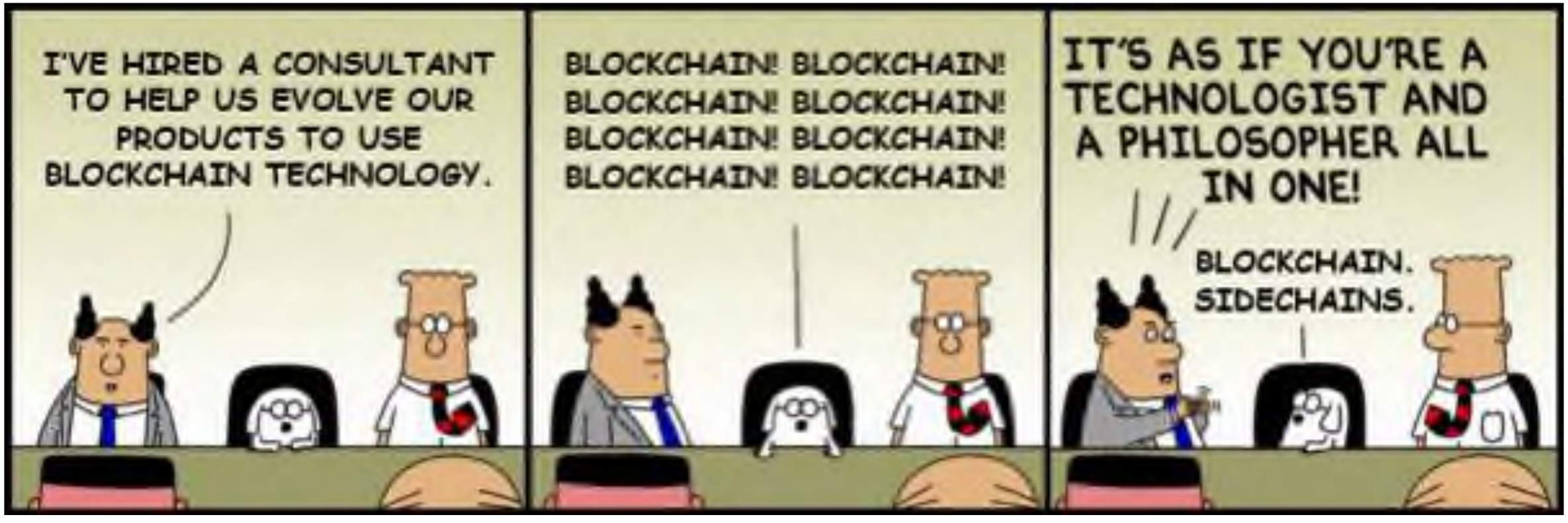
Blockchainization
A first period

http://rmod.lille.inria.fr

Stéphane Ducasse



I THINK WE SHOULD BUILD A BLOCKCHAIN
UH-OH
S Adams E-mail: SCOTTADAMS@AOL.COM
DOES HE UNDERSTAND WHAT HE SAID OR IS IT SOMETHING HE SAW IN A TRADE MAGAZINE AD?
WHAT COLOR DO YOU WANT THAT BLOCKCHAIN?
I THINK MAUVE HAS THE MOST RAM.
© 1995 United Feature Syndicate, Inc.(NYC)


I'VE HIRED A CONSULTANT TO HELP US EVOLVE OUR PRODUCTS TO USE BLOCKCHAIN TECHNOLOGY.
BLOCKCHAIN! BLOCKCHAIN!
BLOCKCHAIN! BLOCKCHAIN!
BLOCKCHAIN! BLOCKCHAIN!
BLOCKCHAIN! BLOCKCHAIN!
IT'S AS IF YOU'RE A TECHNOLOGIST AND A PHILOSOPHER ALL IN ONE!
BLOCKCHAIN. SIDECHAINS.

A word of presentation

Expert in language design

Traits in Php7.0, Perl60
Influenced Traits in Scala
Pharo

Expert in Software Evolution

Program analysis, metamodels, visualization, testing, refactorings, metrics, smart contracts….

Roadmap

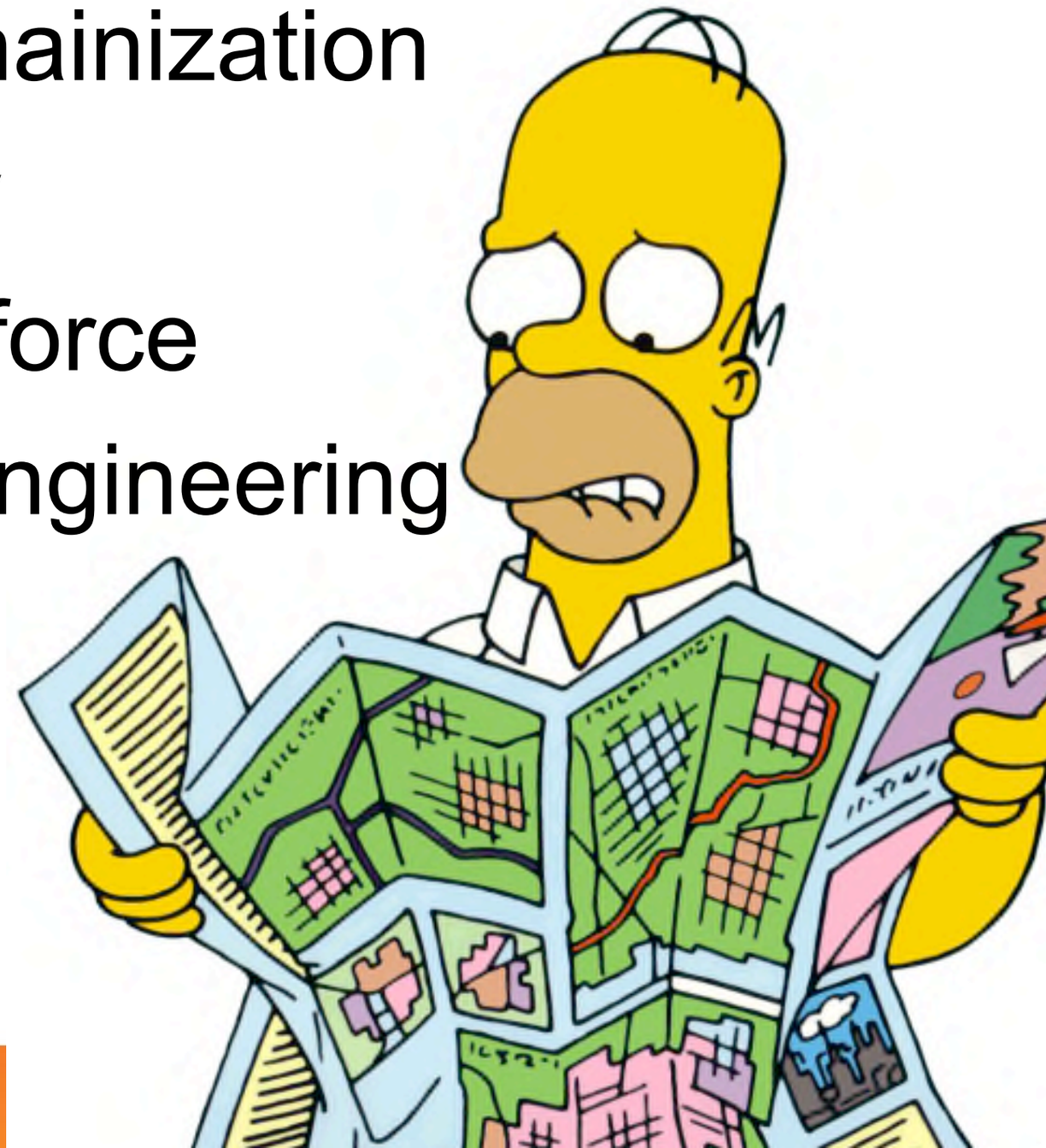
Blockchain in 3 slides

Challenges for Blockchainization

The real face of legacy

How we fight the dark force

Blockchain Software Engineering

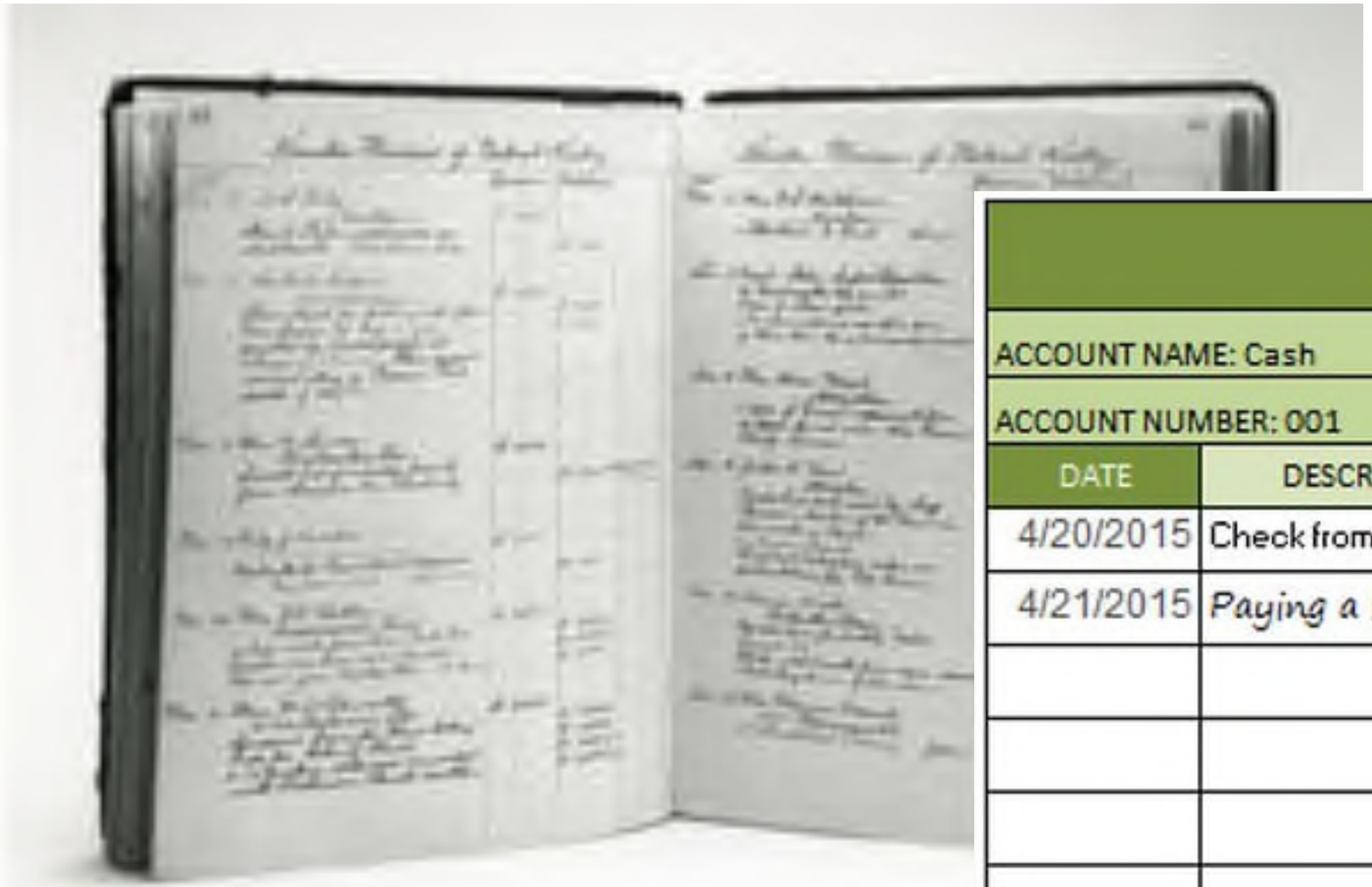


Blockchain…

the new graal

What is Blockchain?
From an User's Perspective

A secured open distributed ledger that promotes privacy



GENERAL LEDGER

ACCOUNT NAME: Cash

ACCOUNT NUMBER: 001

DATE	DESCRIPTION	JOURNAL #	DEBIT	CREDIT	BALANCE
4/20/2015	Check from friend	J1	$ 500.00		$500.00
4/21/2015	*Paying a friend*	J1		$ 200.00	$300.00

What is Blockchain?
From an User's Perspective

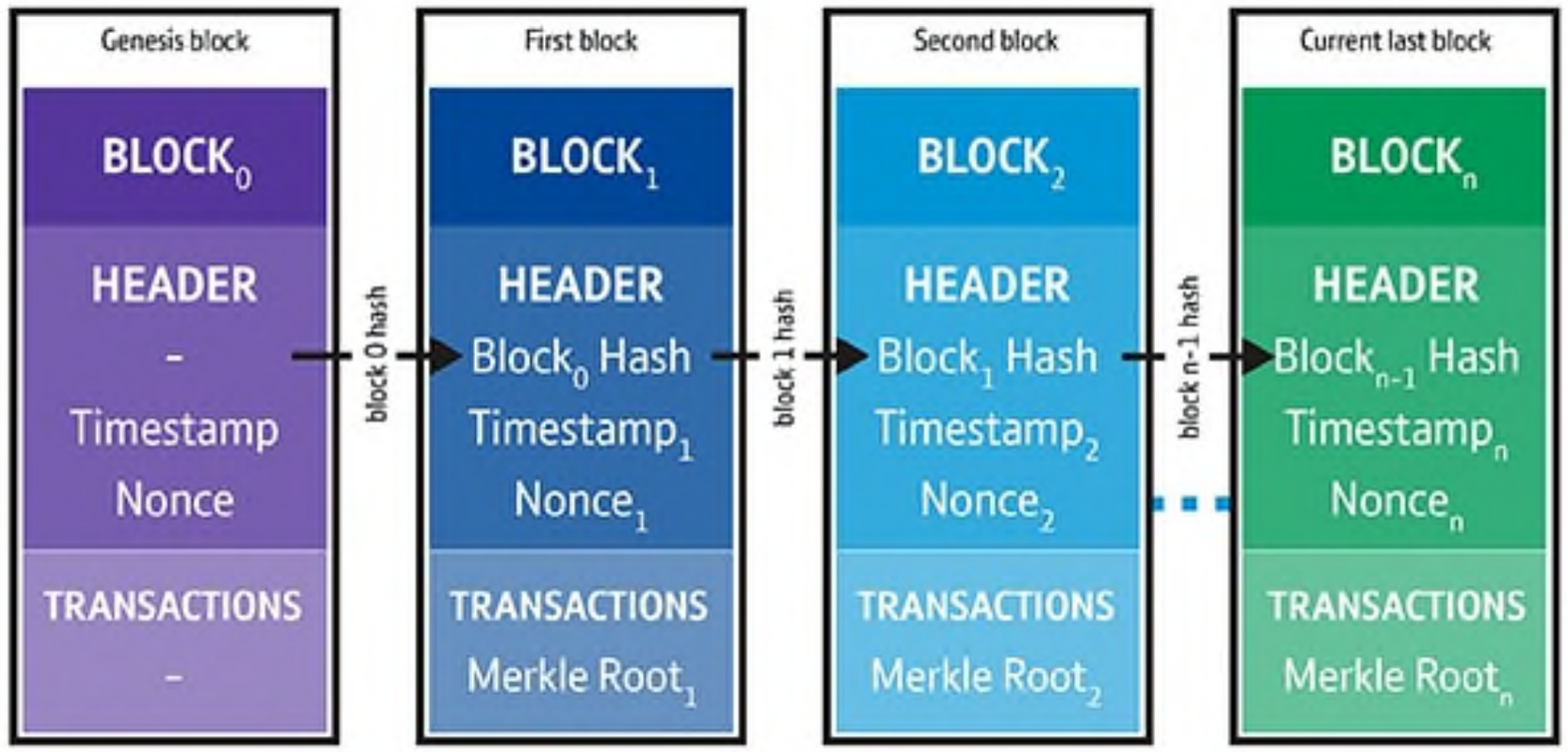
An open distributed Ledger that promotes privacy

- Ledger - Records economic transactions of cryptocurrency (BitCoin, Ether).
- Open - anyone can participate on and inspect the transactions
- Distributed - the ledger is shared and synchronised among several "people"
- Privacy - the ledger transactions promote anonymity

What is Blockchain?

From a Developer's Perspective

Roughly, an append-only globally shared transactional database



What is Blockchain?
From a Developer's Perspective

Roughly, an append-only globally shared transactional database

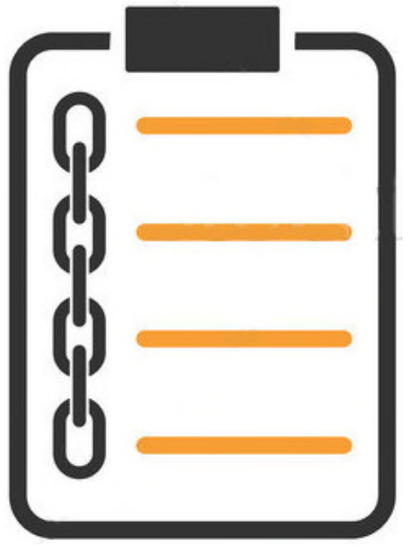
- Database — stores and manages digital resources
- Transaction — ACID properties
- Append-only — immutable data (cannot delete or alter), just append new information
- Globally Shared — managed by a P2P network, all peers store a complete copy of the database

Smart Contracts
[Fritz Henglein, Smart Contracts…, 2017]

"Smart Contracts are neither Smart nor Contracts"

- Smart Contracts are self-executing programs coded in complex Turing-complete language.
- Rules & Actions intermixed
  (not really a contract)
- Low-level programs hard to analyze
  (not really smart)

Smart Contracts
[Henrique Rocha, CBSoft, 2018]



Smart Contracts are objects deployed in the blockchain.

- Code and data (like a class) residing at a Blockchain address.
- Transactions created by function execution.
- Contracts are sandboxed in the Blockchain.

Challenges for blockchainisation

Data

Flow

Architecture

Architecture of trust

At what cost?

Challenges … data

How data created/manipulated/merged?

What part of data stored?

Is the data manipulated/stored till needed?

Where data is stored?

Challenges … flow

How data flows between subcomponents/subsystems?

How to capture existing process to hook into?

Challenges … trust

What are the trustees?

What is the impact of trust on the architecture (off-chain/on-chain)?

The Real Faces of Legacy

Lack of Knowledge

- ***Obsolete*** or ***no*** documentation
- ***Departure*** of the original developers or users
- Disappearance of ***inside*** knowledge about the system
- ***Limited*** understanding of entire system
- ***Missing*** tests
- ***No*** versioning

Code symptoms

- Duplicated code (up to 60% on real systems)
- Dead code (up to 2/3 on real systems)
- Code smells
- Big build times

Process symptoms

- ***Too long*** to turn things to production
- Need for ***constant*** bug fixes
- Maintenance dependencies
- Difficulties separating products
- Simple changes take too long

Some true examples

- Company C has a super large VB application. No idea of the use of underlying windows API. Can they migrate incrementally?
- Company B has 60% code duplication of Cobol. The single developer left to competitor. No tests. 65 M CHF to replace the system (SAP).

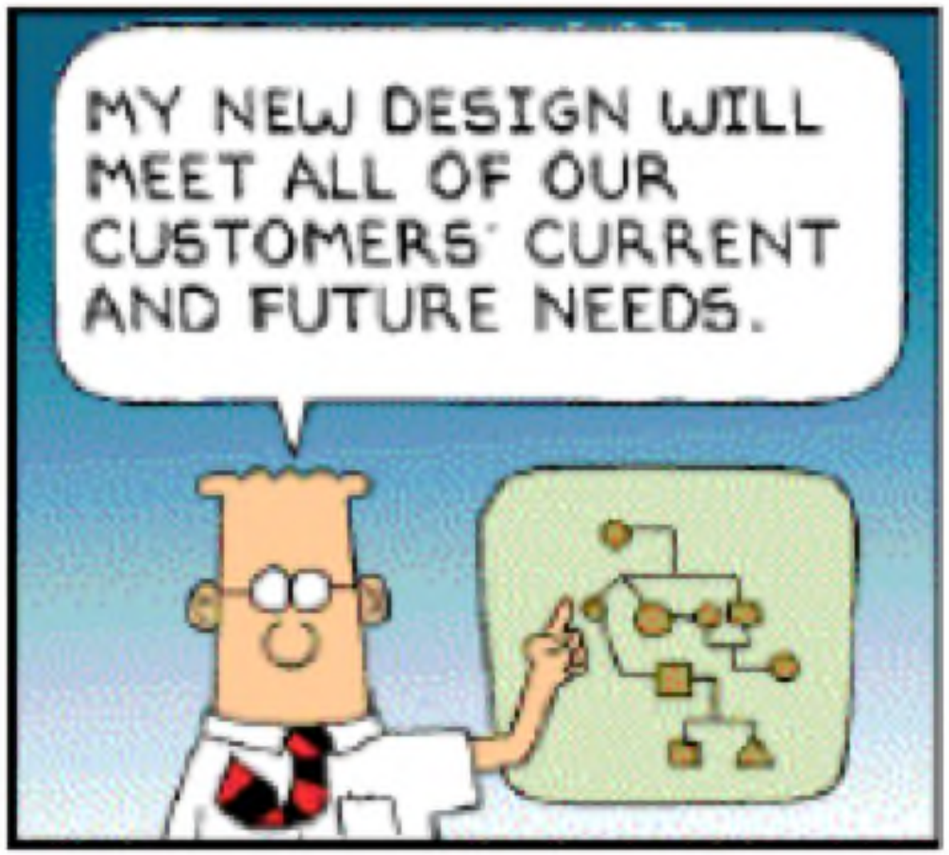
Some true examples

- Company K has 700,000 fine. Legacy is a post cobol system language running half of their business: 4500 programs. No other information.
- Company X has no versioning control, no automated tests, ship to clients new version without patch Y. Pascal + form from 90s.
- Company Z has a large Java6 app with many dependencies on obsolete/unmaintained “solutions”. Hell of dependencies.

Software is Complex

The big ball of mud

An instant of fun



Laws of software evolution

Continuing change

A program that is used in a real-world environment must change, or become progressively less useful in that environment.

Increasing complexity

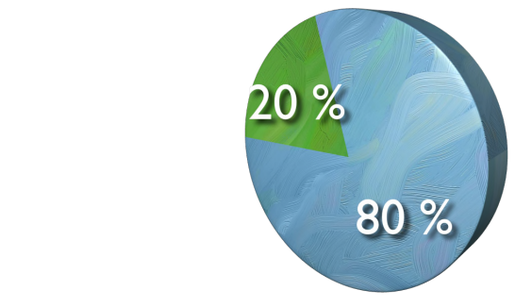
As a program evolves, it becomes more complex, and extra resources are needed to preserve and simplify its structure.

Software is a living entity...

- Early decisions were certainly good at that time
- But the context changes
- Customers change
- Technology changes
- People change

We only maintain useful successful software

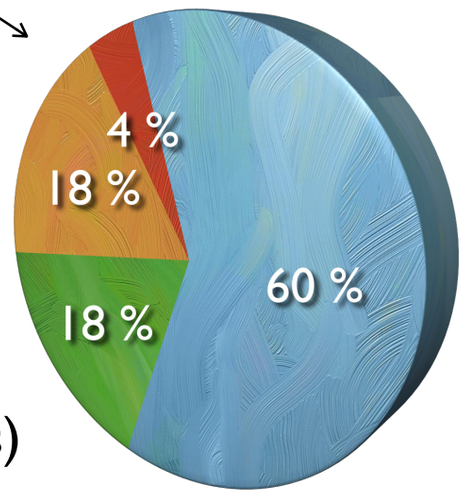
Maintenance is *continuous* Development



Between **70**% and **90**% of ***global*** effort is spent on "maintenance" !

18% Adaptive
(new platforms or OS)

18% Corrective
(fixing reported errors)

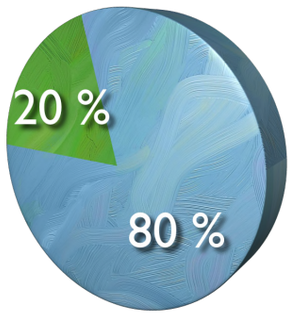


4% Other

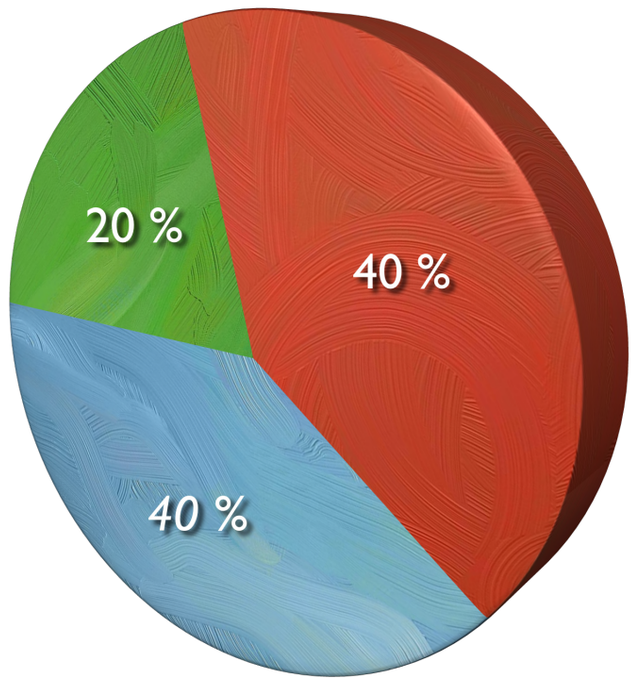
60% Perfective
(new functionality)

"Maintenance"

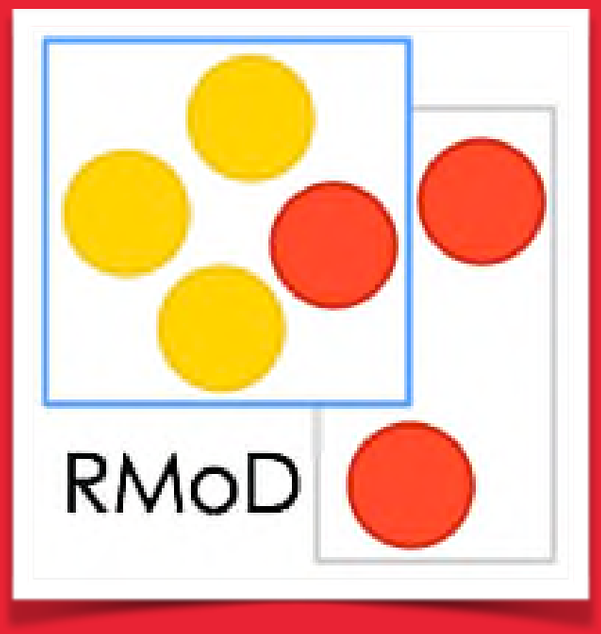
50% of development time is lost trying to understand code !



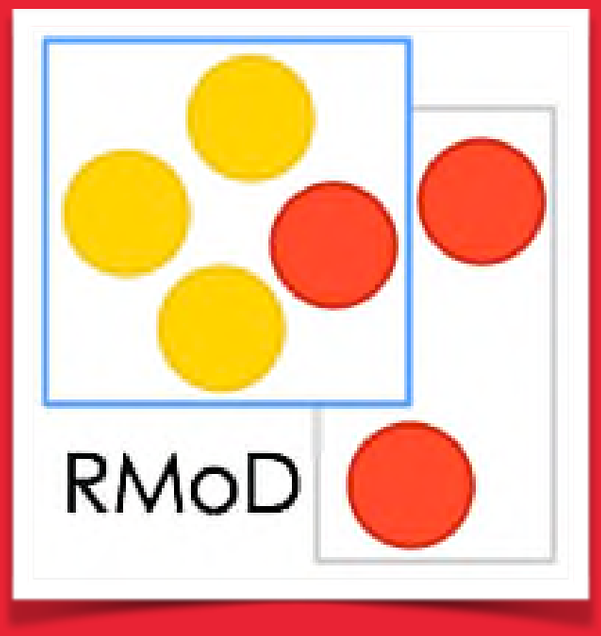
Between **50**% and **80**% of the ***overall cost is spent in the evolution***



We lose a lot of time with inappropriate and ineffective practices



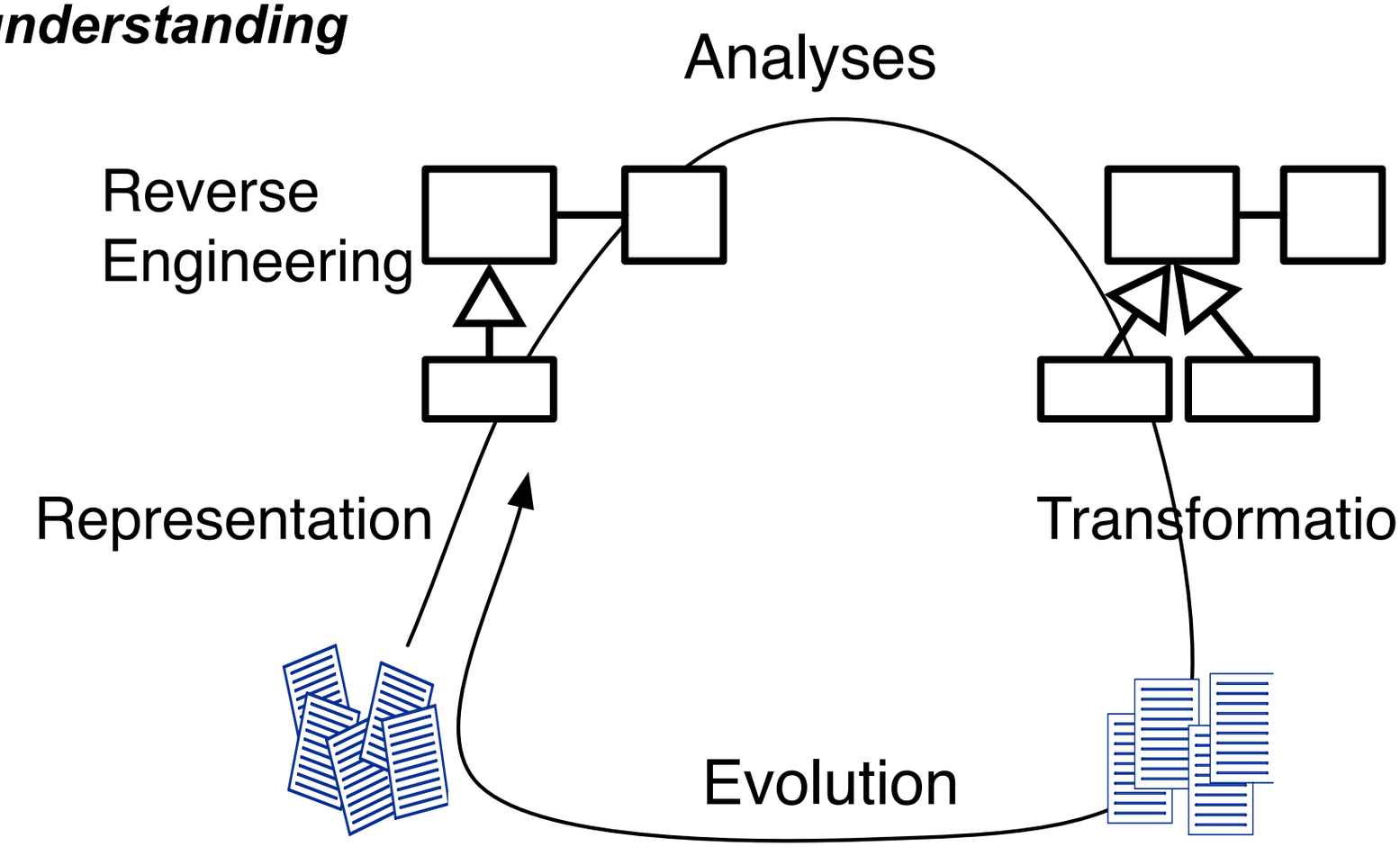
How can we fight the dark force?



We design tools and analyses to tame software

Expertise

code analysis, metamodeling, software metrics, reverse engineering, ***program visualization***, evolution analysis,
refactorings, quality, ***program understanding***
changes analysis, commit,
dependencies,
rule and bug assessment
semi-automatic migration
example-based transformations
test selection, rearchitecturing
blockchains, ***ui-migration***

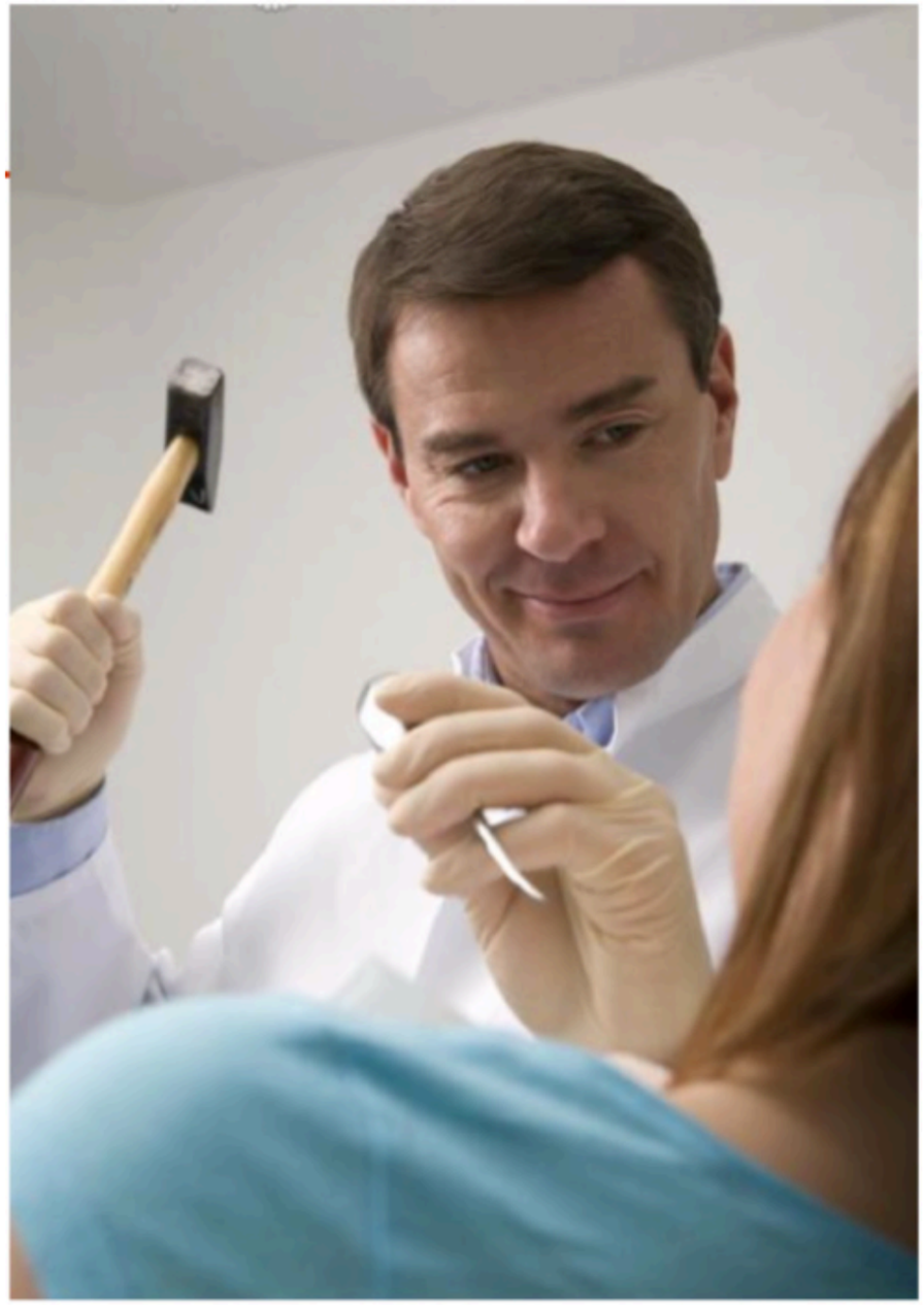


Collaborations

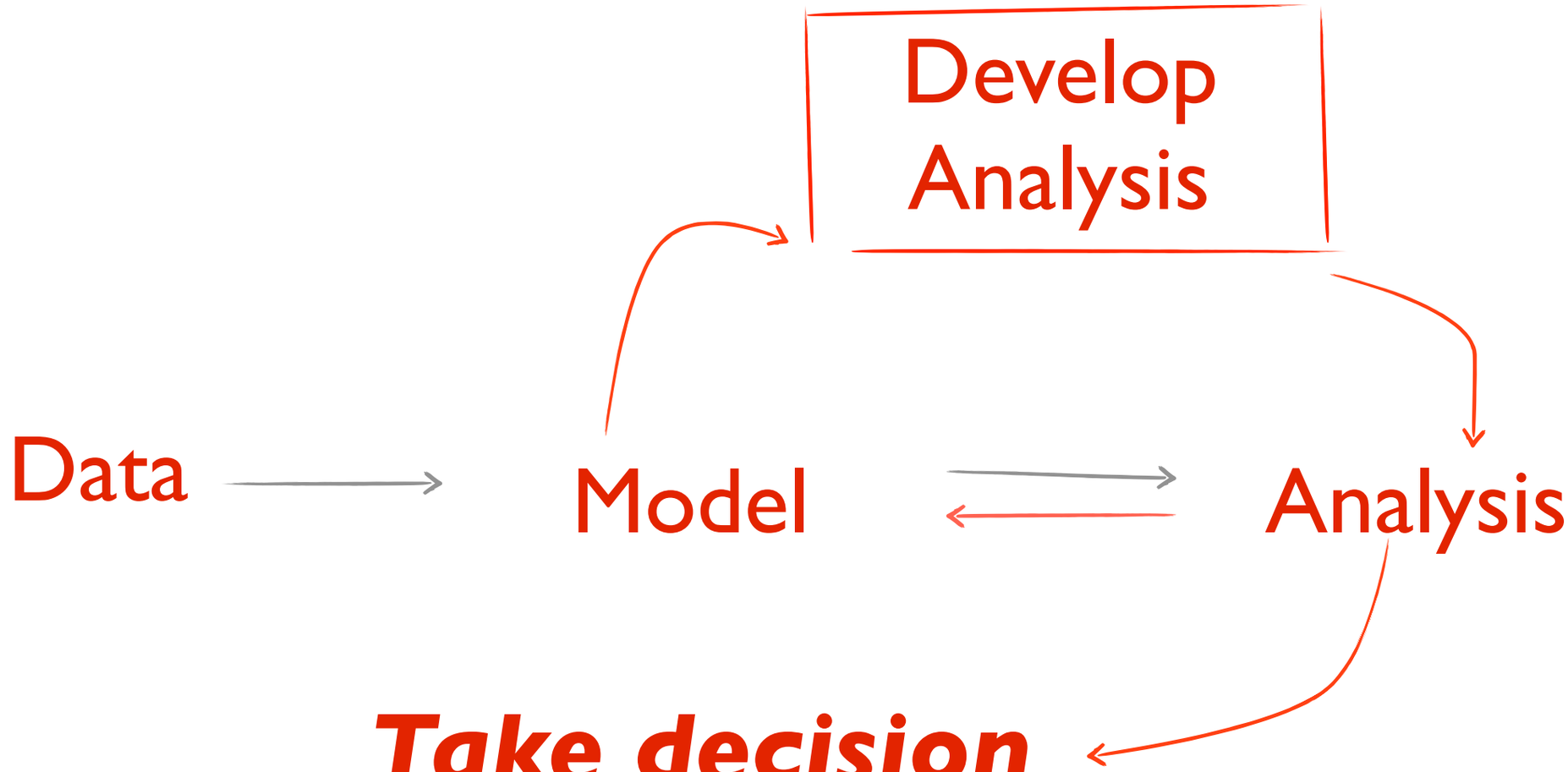
Pleiad (Chile), UFMG (Brazil),
SCG (Switzerland), Soft-VUB (Belgium)

You would not see this dentist!

Why doing it for your software?

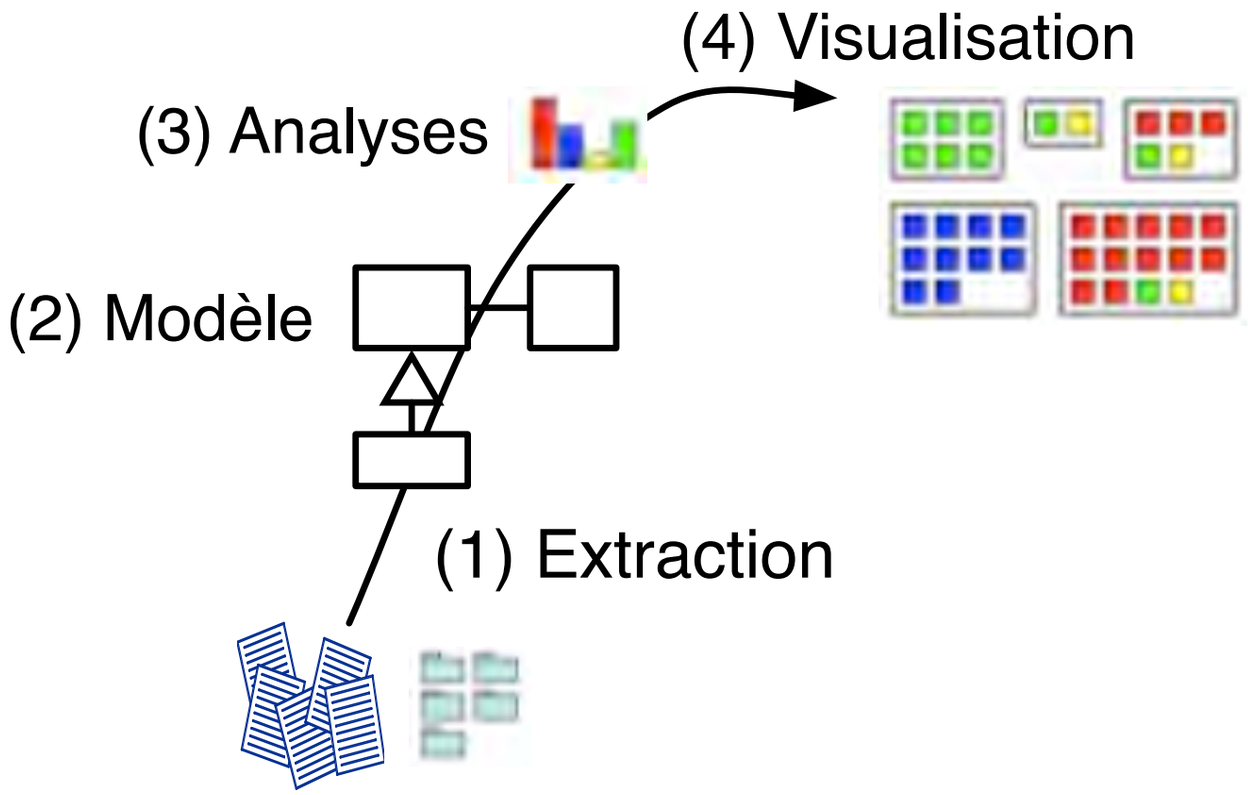


Building dedicated tools

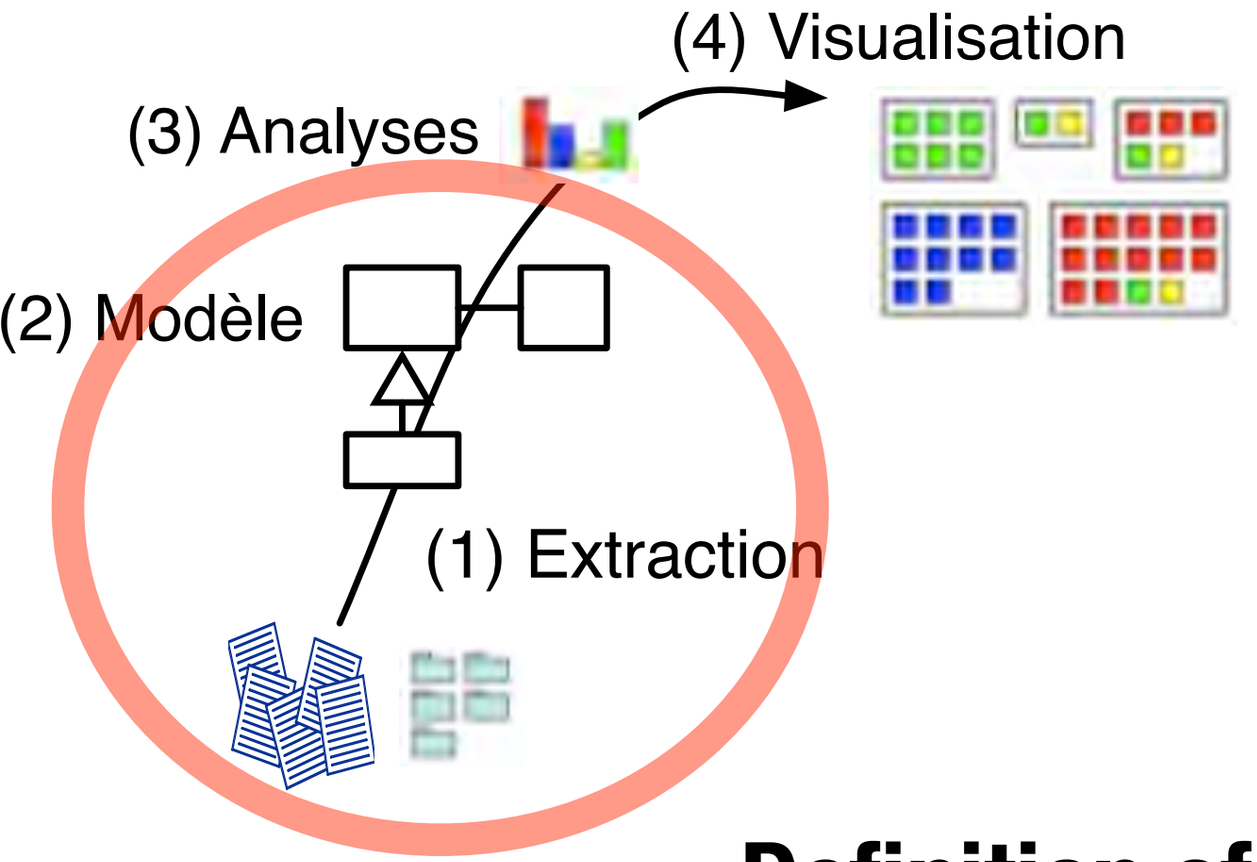


an analysis should lead to a decision

Example : Who is behind package X ?



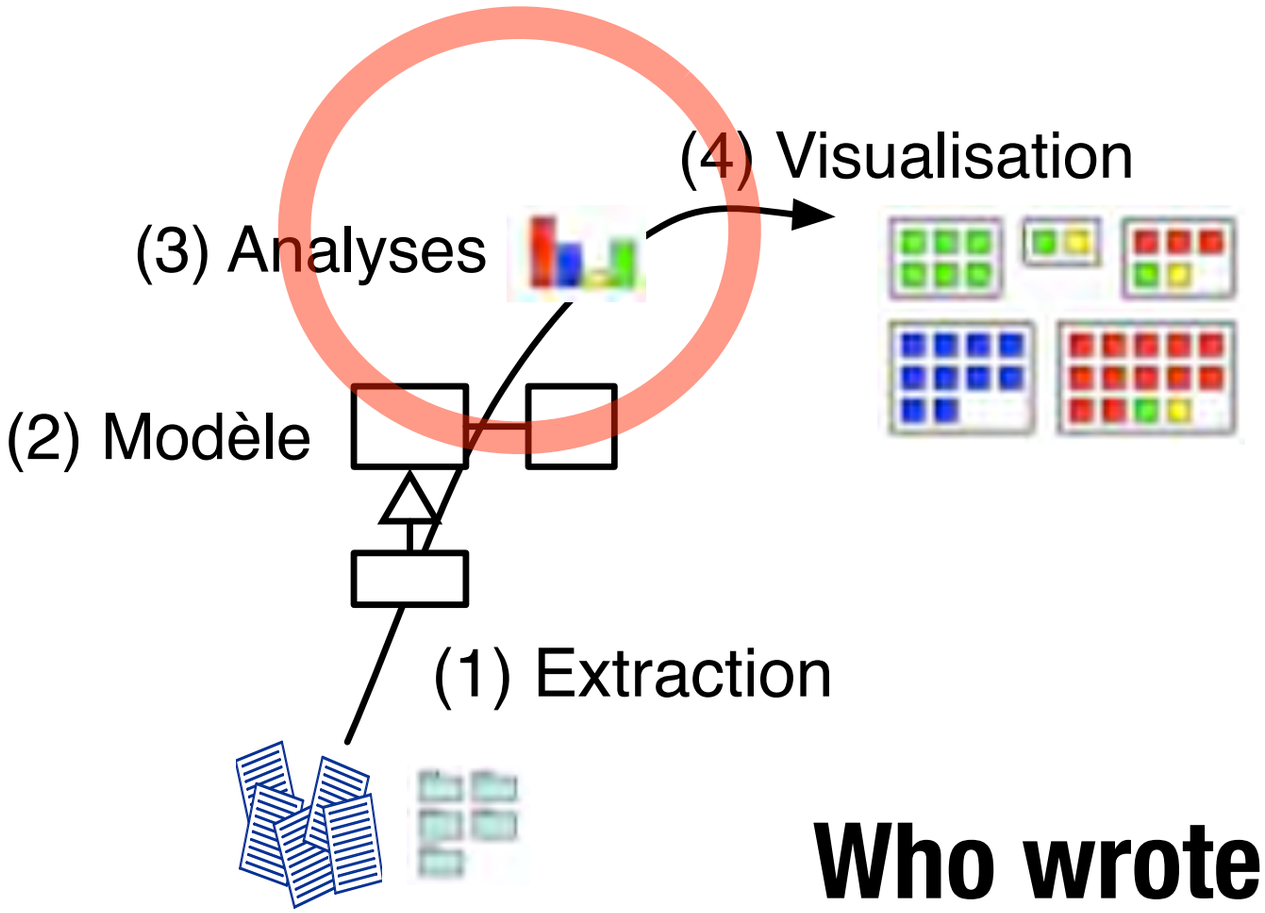
Step 1 - Model Creation/Import



Definition of a model to represent entities

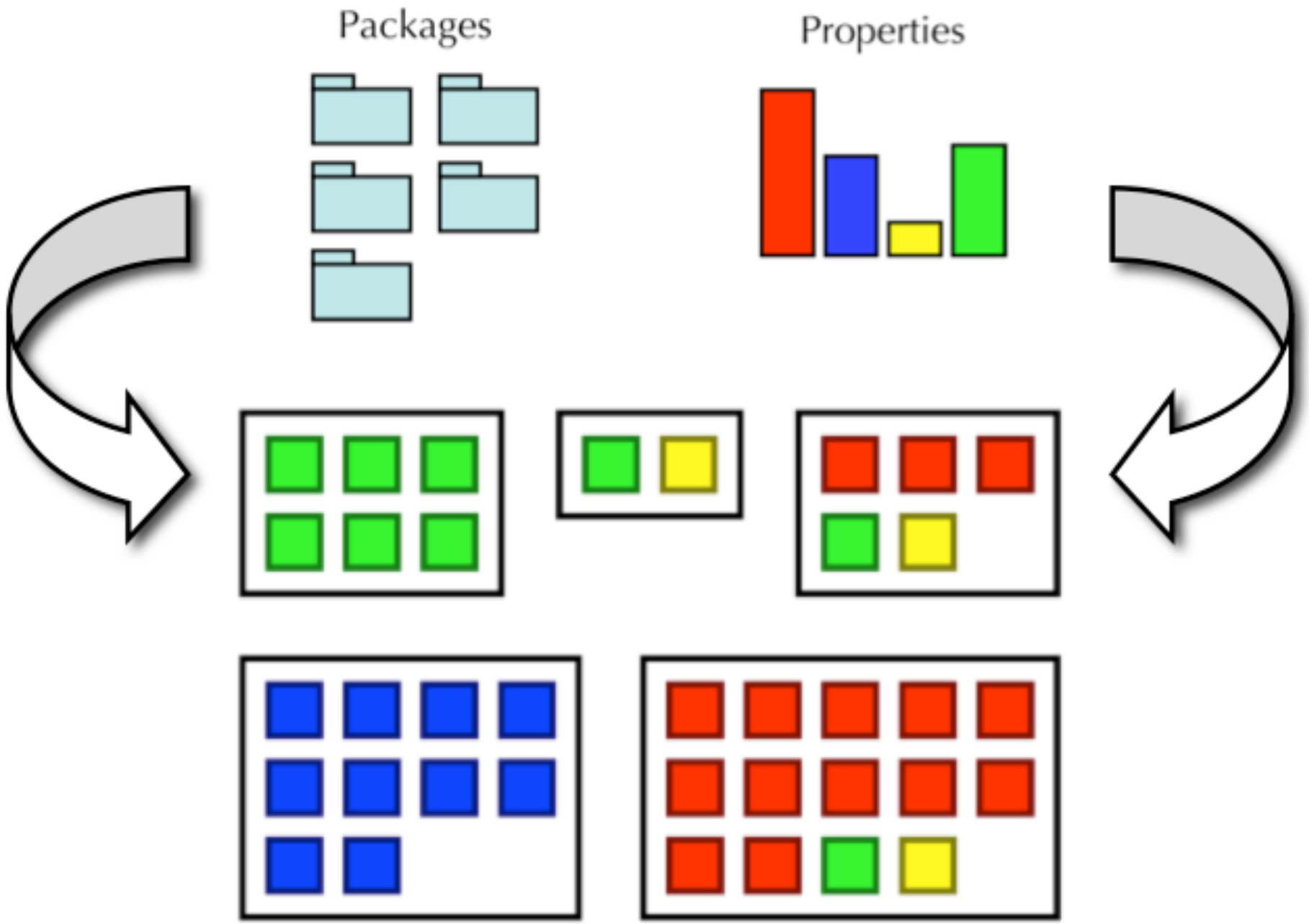
Data Extraction (CVS...)

Step 2 - Analyses



Who wrote how many lines of code?

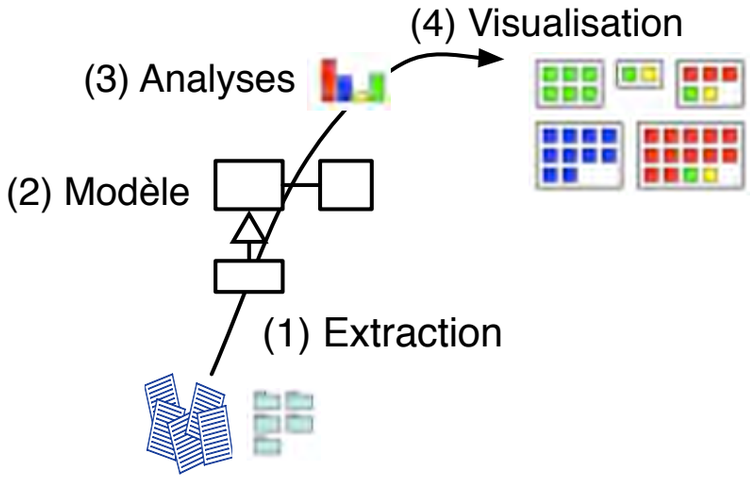
Step : 3 - Creating the Map



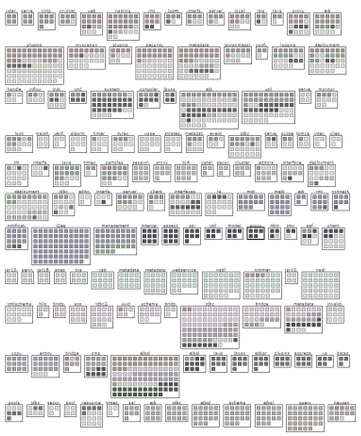
All JBoss at a glance

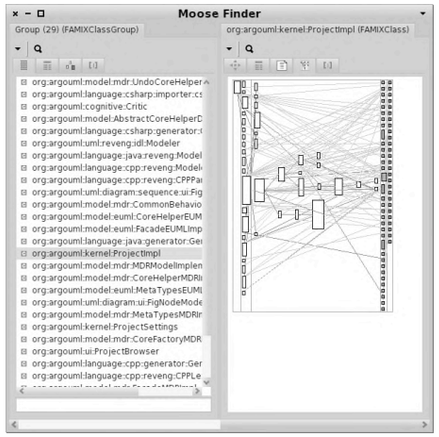
Interactive tool

Data in perspective



A *meta* platform to create tools

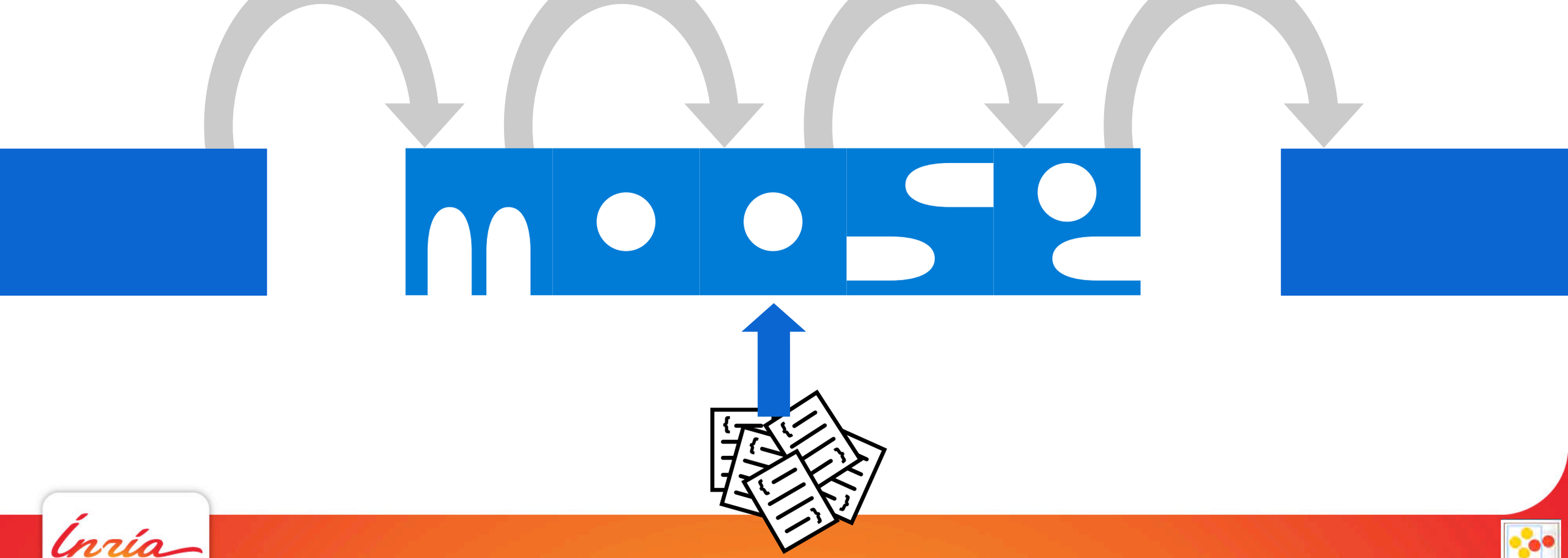


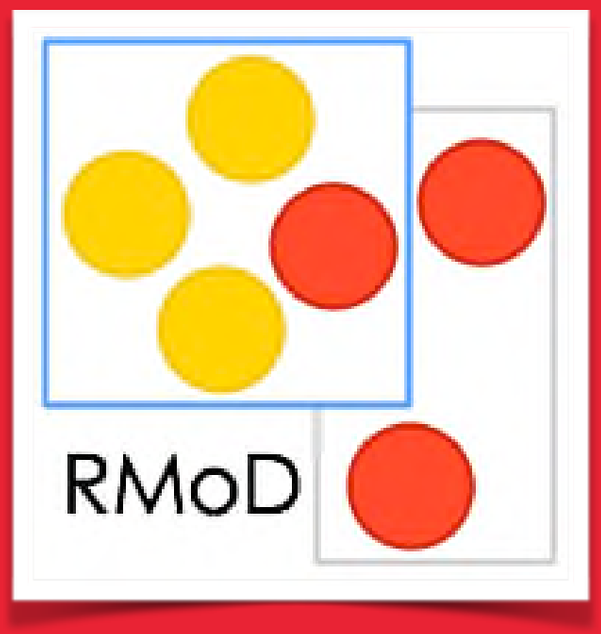


classes select: #isGod

McCabe = 21

LOC = 753,000

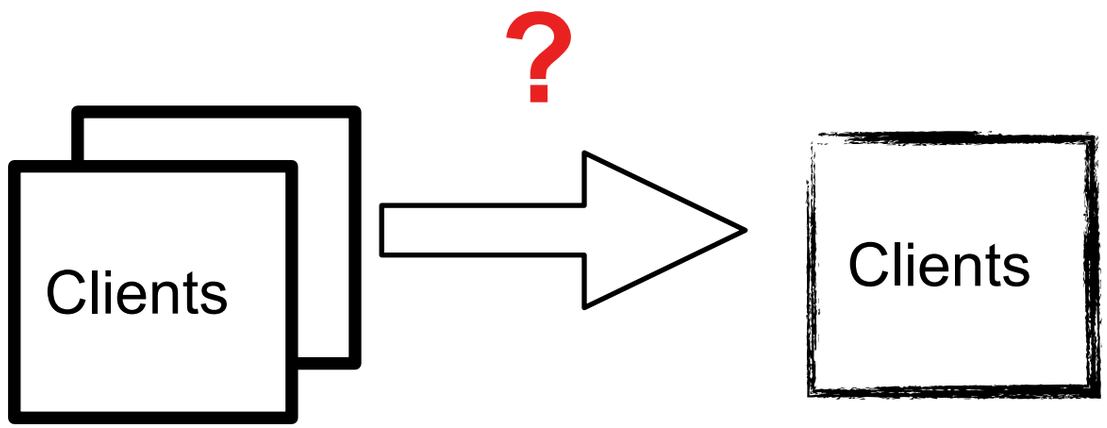


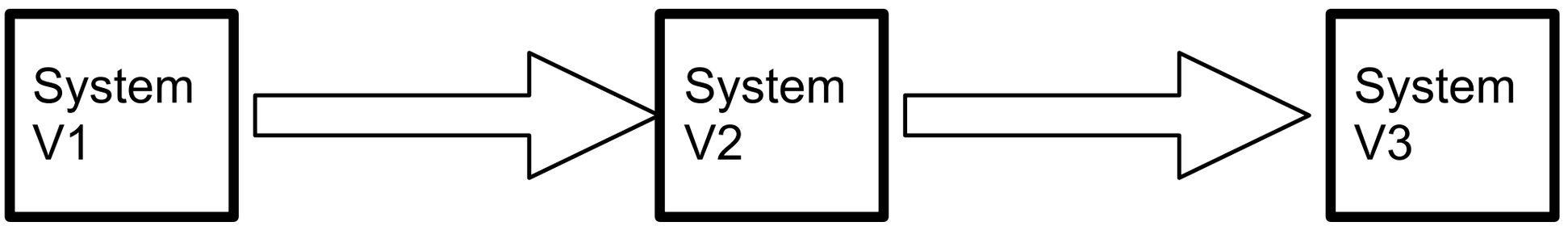


Evolution in the large:

Automated Rules to support migration

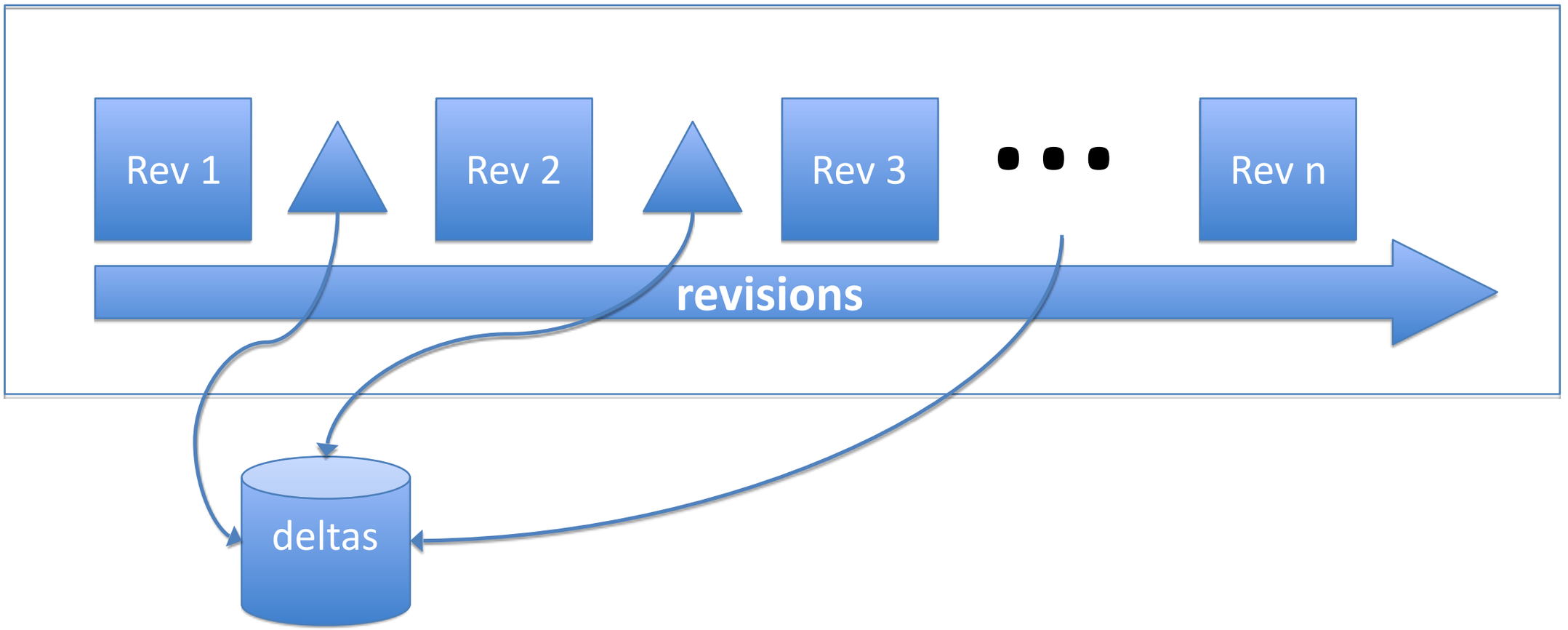
How to help migrate from version to version?





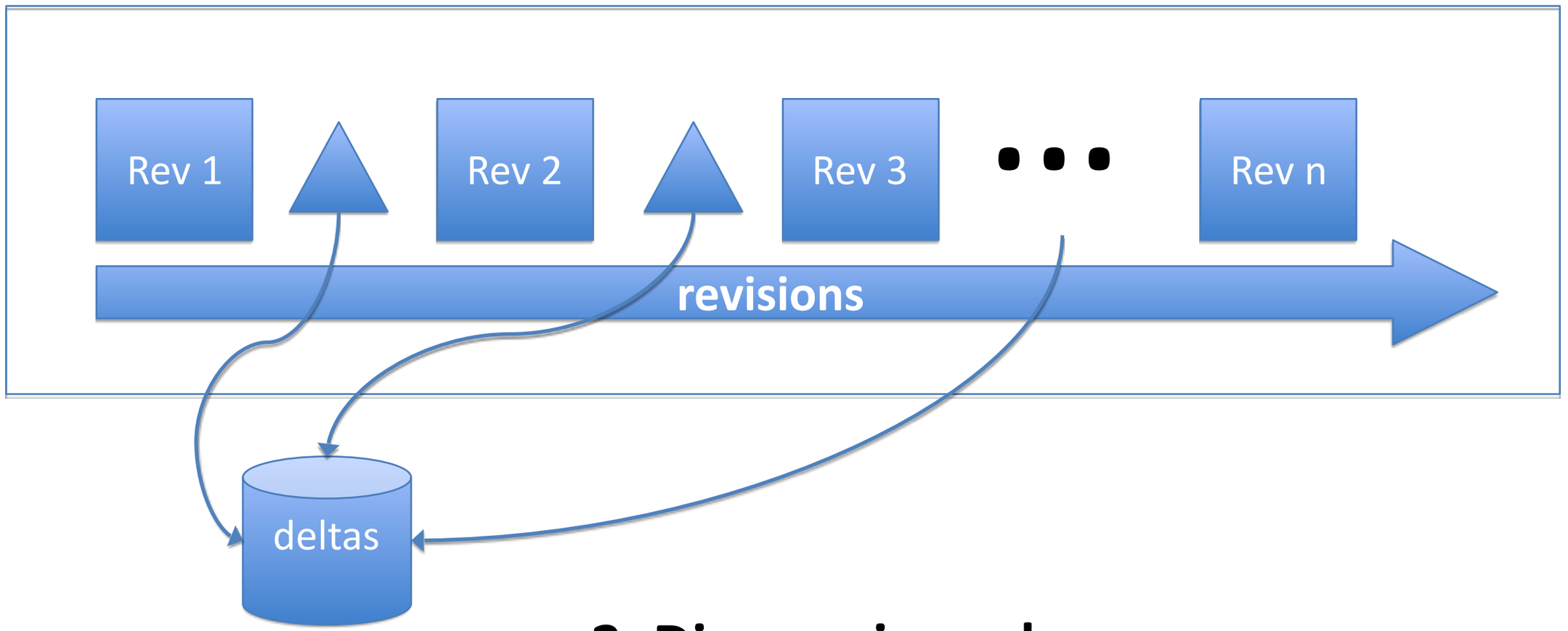
Mining API Change Rules

1. Extracting deltas

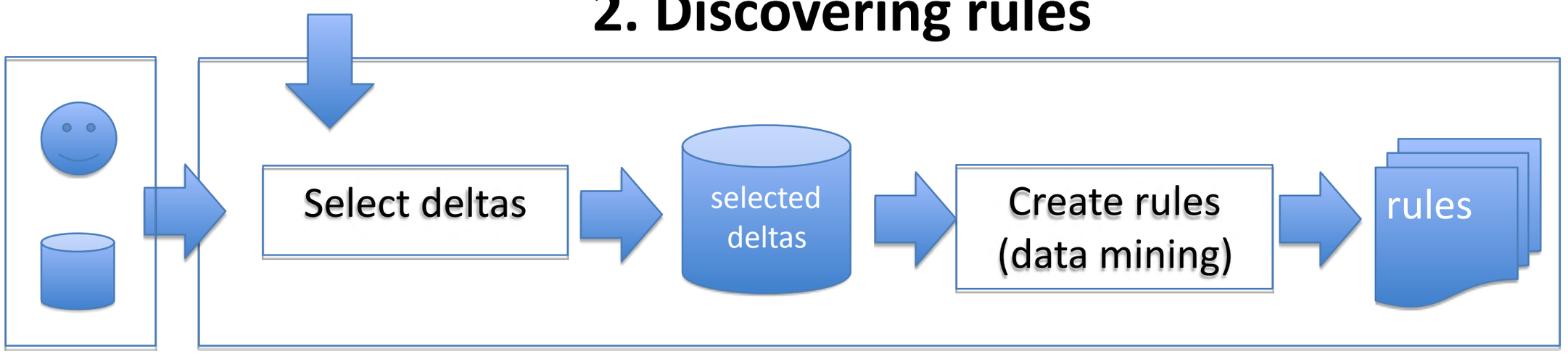


Mining API Change Rules

1. Extracting deltas



2. Discovering rules



Mining API Change Rules

Format of the changes

deleted-invoc(context-id, signature)

added-invoc(context-id, signature)

Diff of method foo() between version 1 and 2

```
– self.add(MooseModel.root().add(model));
+ self.add(model.install());
```

Formatted changes

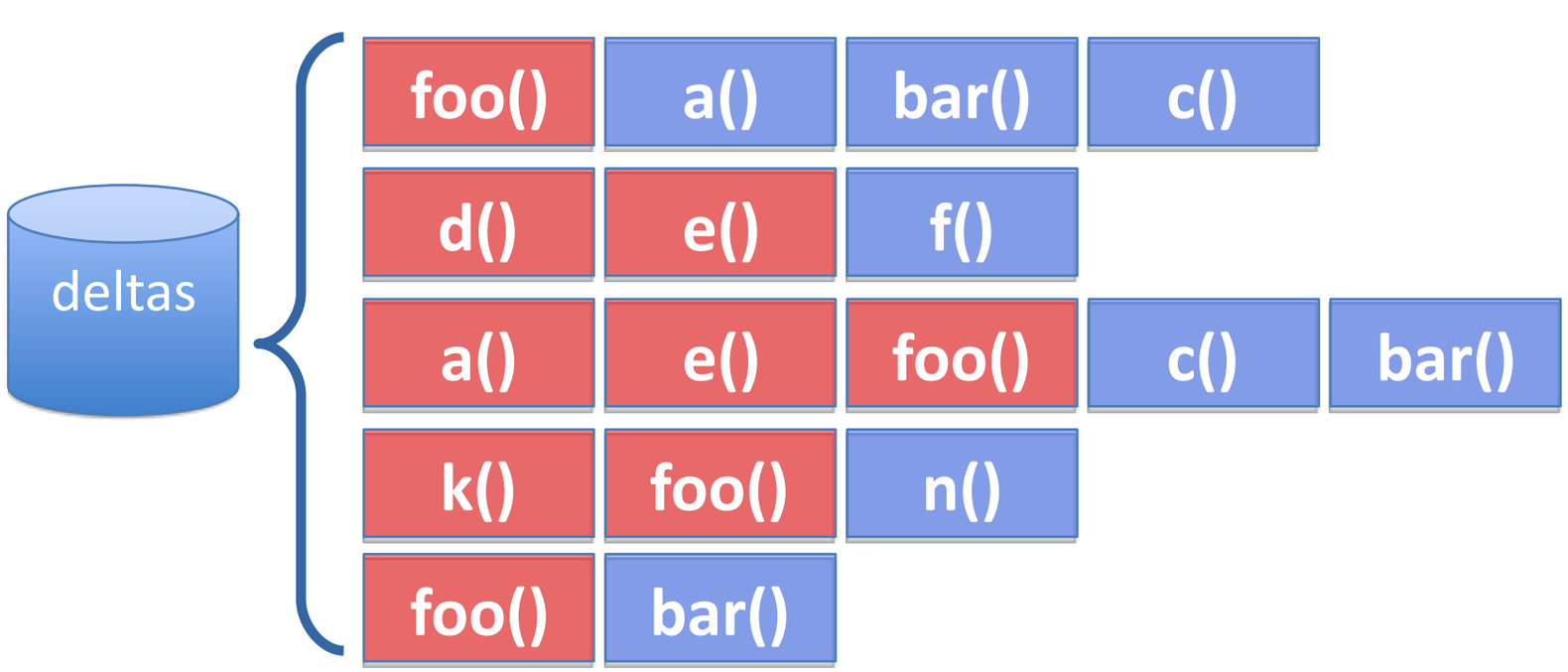
deleted-call(“foo()-rev2”, “MooseModel.root()”)

deleted-call(“foo()-rev2”, “MooseModel.add(MooseModel)”)

added-call(“foo()-rev2”, “MooseModel.install()”)

Mining API Change Rules

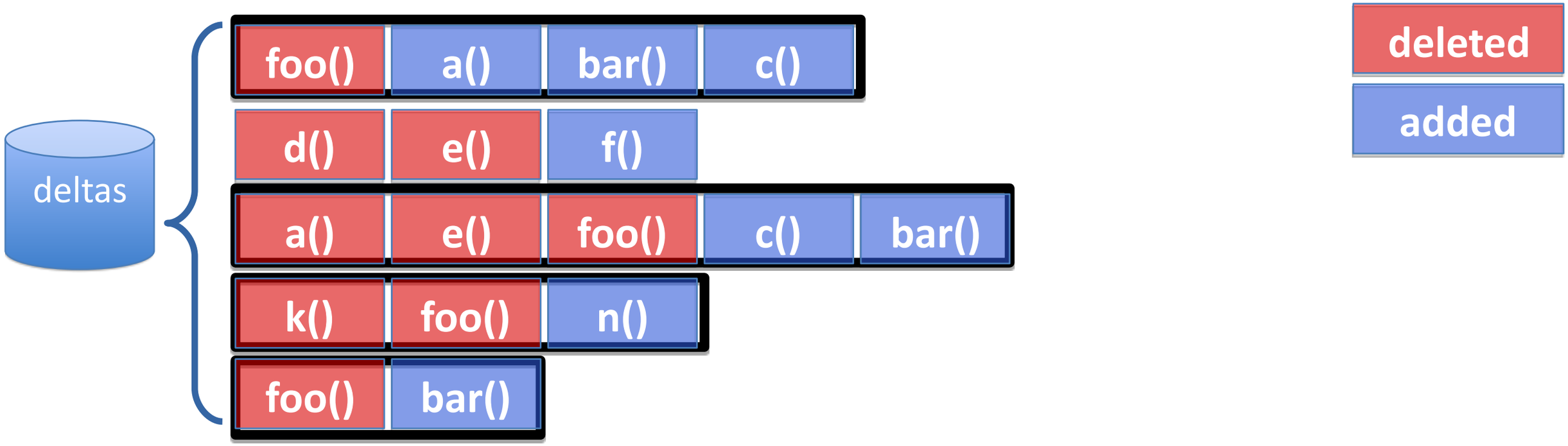
Request: **foo()**



Mining API Change Rules

Request: foo()

1st step: selecting deltas

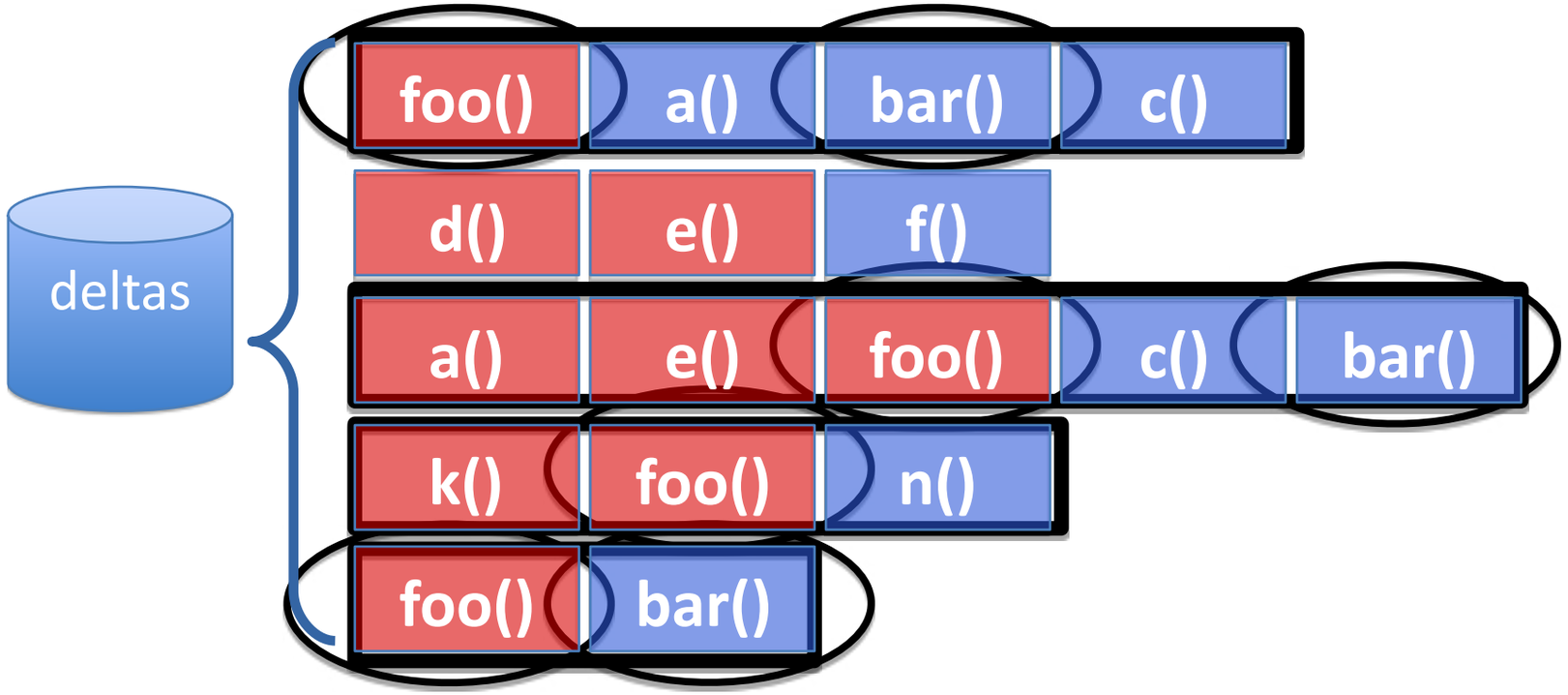


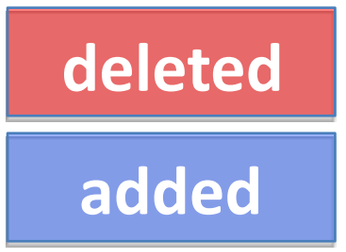
Mining API Change Rules

Rule: foo() → bar() Confidence = 75%

1st step: selecting deltas

2nd step: discovering rules





Mining API Change Rules

isNil().ifTrue(*) → ifNil(*)

keys().do(*) → keysDo(*)

intersect(*) → intersectIfNone(*,*)

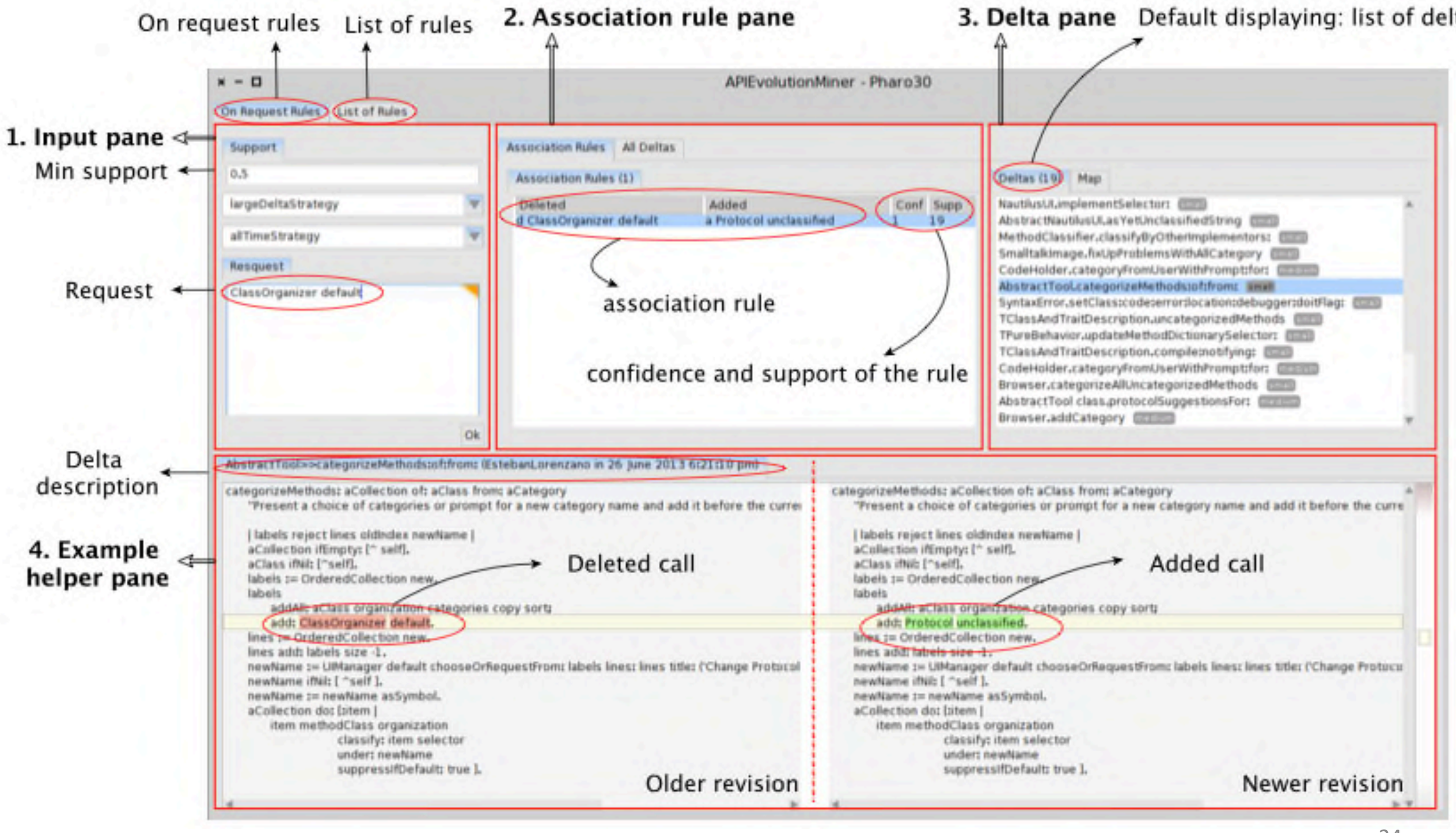
Scanner.new().scanTokens(*) → parseAsLiteralToken()

RegisterAsApplication(*) →

WAAdmin.registerAsApplicationAt(*,*)

Character.cr() → ROPlatform.current().newLine()

Tool Support

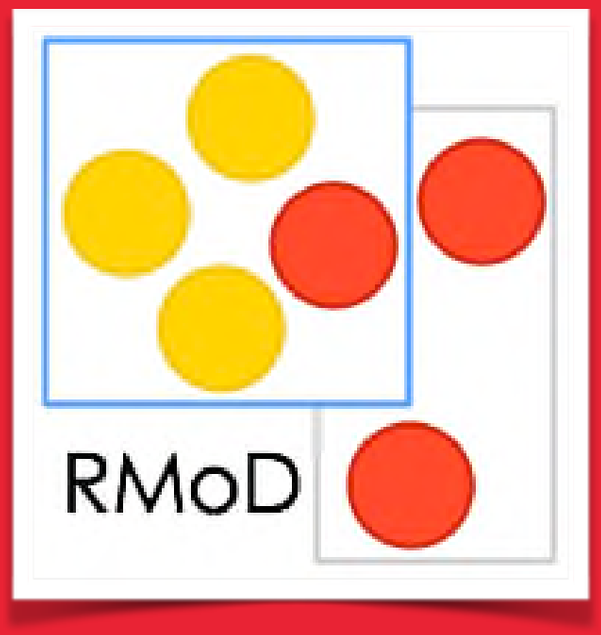


Validation

Are the generated rules valid to experts?

Yes between 46% to 86%

Rules	Valid	Invalid	Don't know	Precision
Pharo2	62	17	20	25	**46%**
Pharo3	95	68	24	3	74%
Moose	50	43	7	0	**86%**
Glamour	23	15	8	0	65%
Roassal	36	28	8	0	78%
Seaside	76	61	15	-	80%



Evolution in the large:

GWT to angular

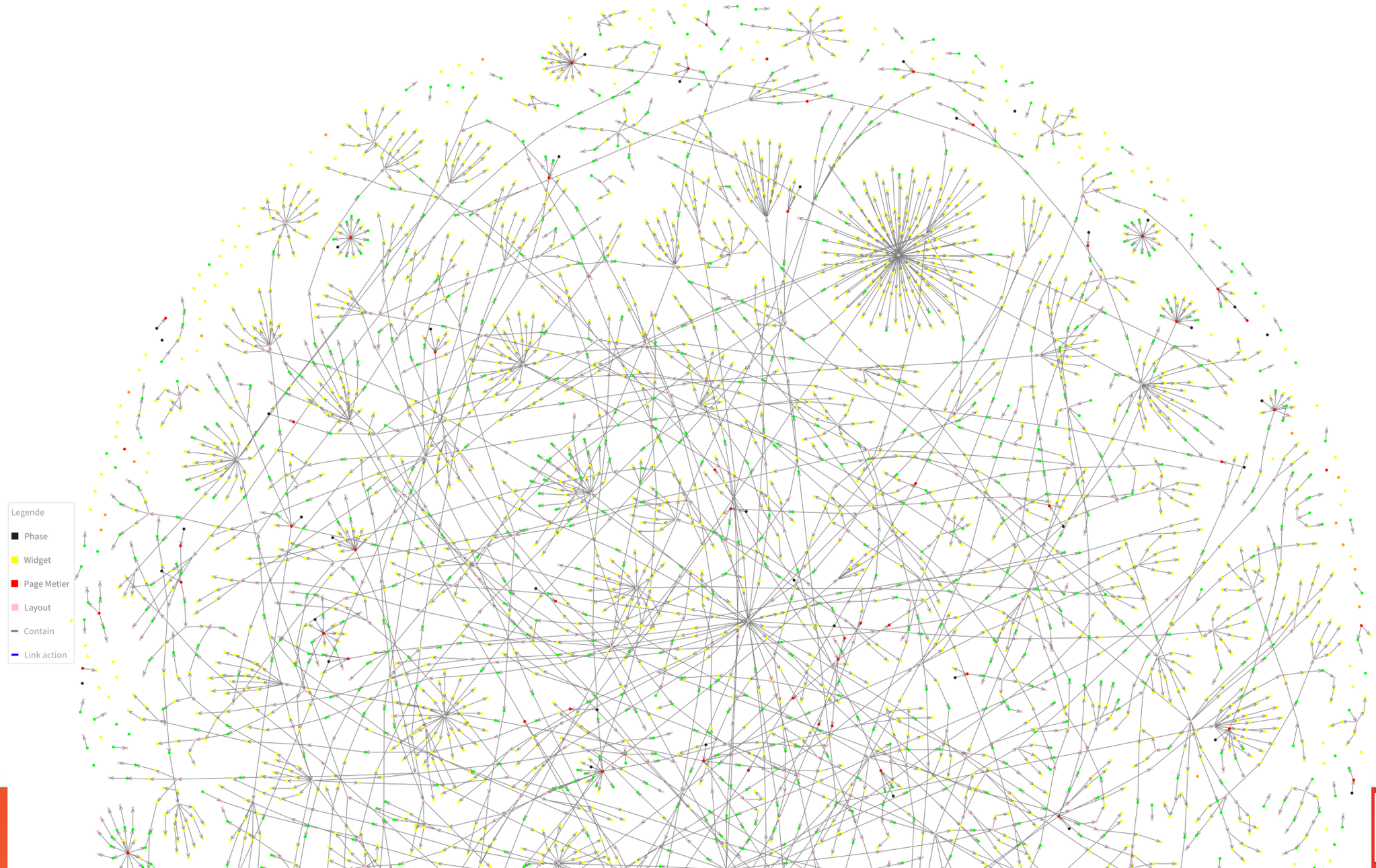
PhD CIFRE of Benoit Verhaeghe for Berger-Levrault

GWT: An industrial Case Application

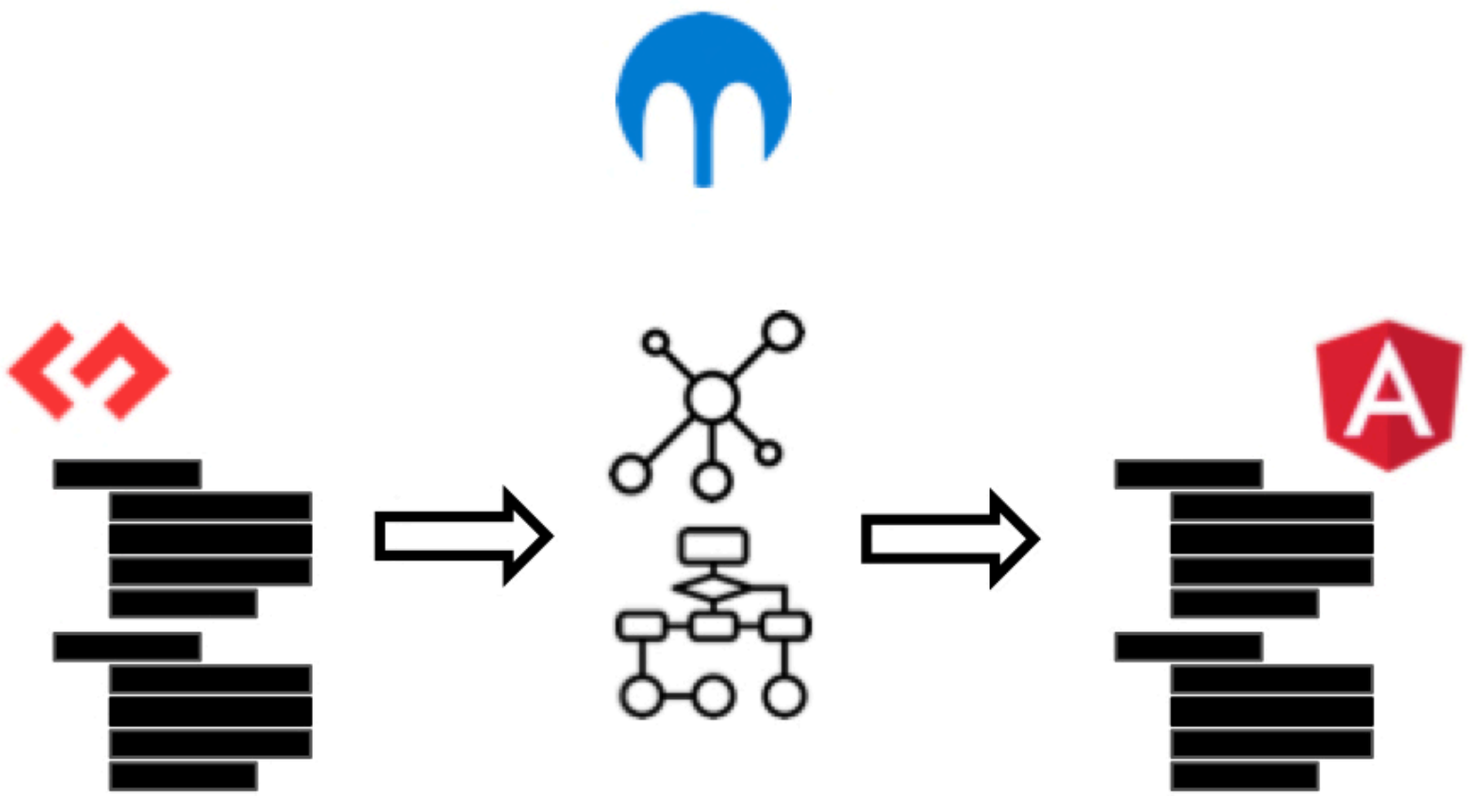
- 26 000 classes
- 450 web pages
- 974 000 invocations
- 1 063 163 LOC (UI)

GWT little bac a sable

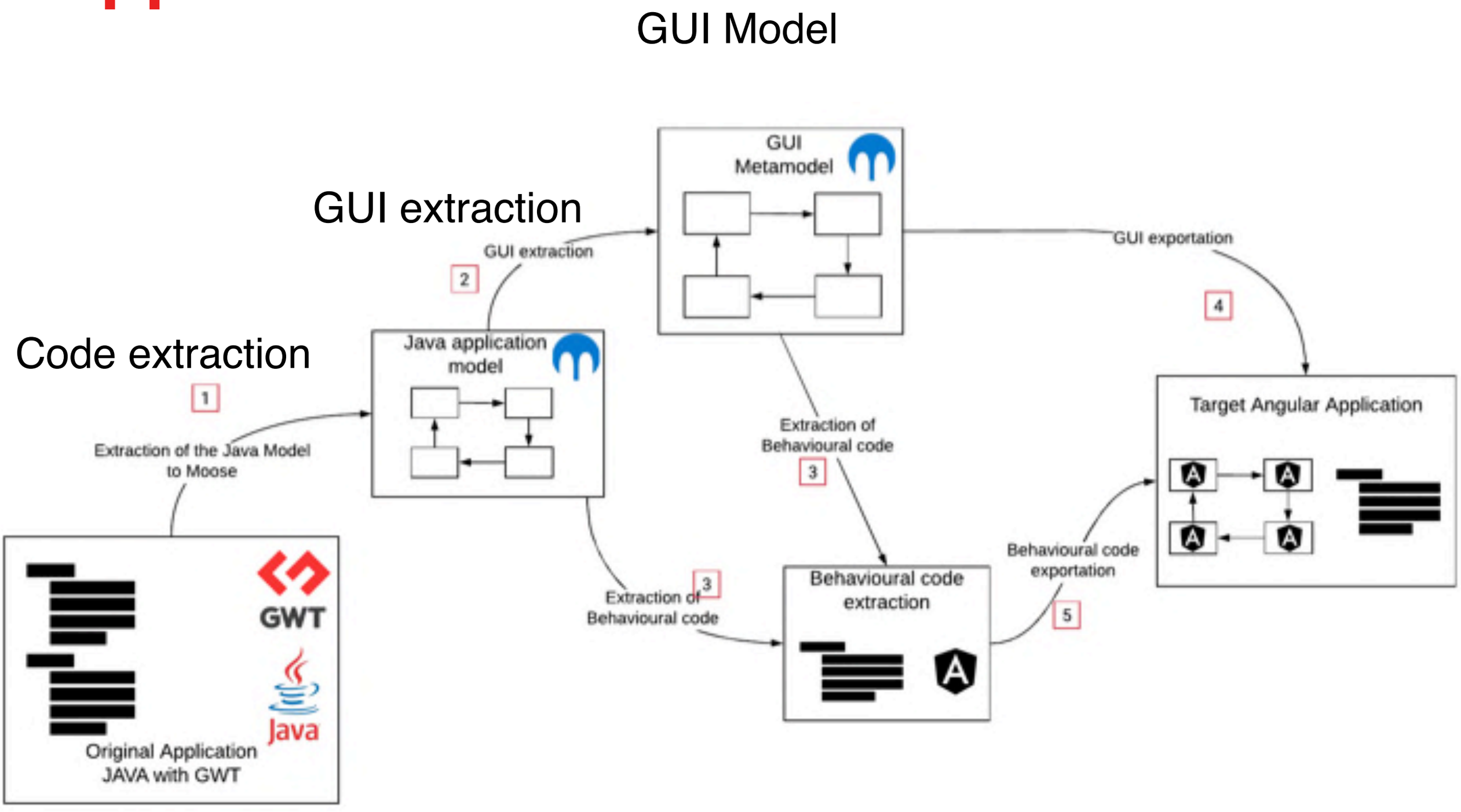
bigKitchen ~ 8979 éléments



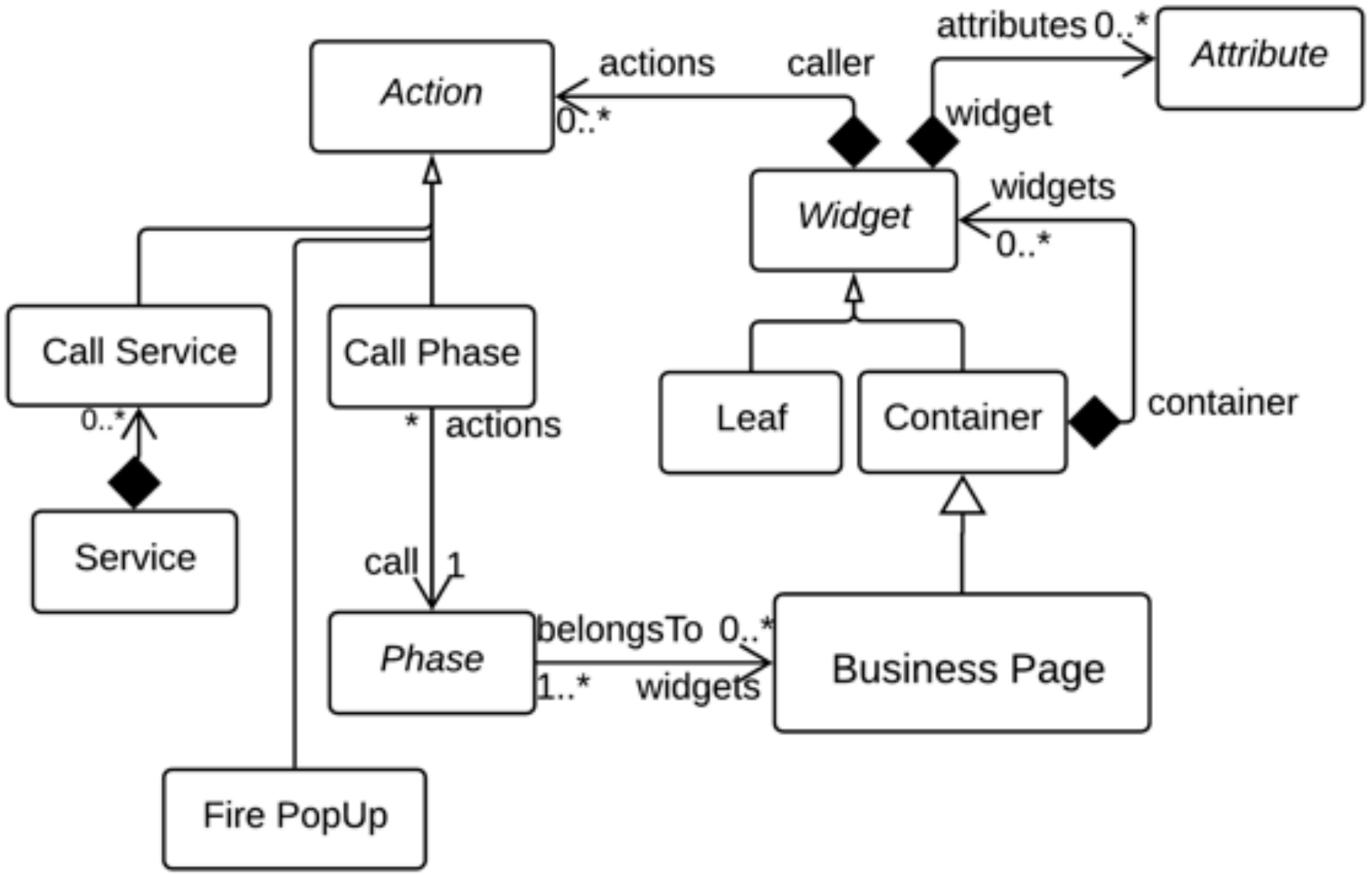
Approach



Approach


GUI Model
GUI extraction
Code extraction
GUI Metamodel
GUI extraction
2
GUI exportation
4
Java application model
1
Extraction of the Java Model to Moose
Extraction of Behavioural code
3
Target Angular Application
Extraction of Behavioural code
3
Behavioural code extraction
Behavioural code exportation
5
GWT
Java
Original Application JAVA with GWT

Metamodel


Action
actions
0..*
caller
attributes 0..*
Attribute
widget
Widget
widgets
0..*
container
Call Service
0..*
Service
Call Phase
* actions
call 1
Phase
belongsTo 0..*
1..* widgets
Leaf
Container
Business Page
Fire PopUp

Results

Export

Navigation par phase

- Ouvrir un onglet
- Ouvir une boite de dialogue modale
- Ouvrir une boite de dialogue non modale
- Ouvrir une boite de dialogue unique non modale

Boites de dialogue (hors phase)

- Boite de dialogue modale
- Boite de dialogue non modale
- Message d'information
- Message d'avertissement
- Message d'erreur (métier)
- Message de confirmation
- Message d'erreur (exception)

Navigation par phase

- Ouvrir un onglet
- Ouvir une boite de dialogue modale
- Ouvrir une boite de dialogue non modale
- Ouvrir une boite de dialogue unique non modale

Boites de dialogue (hors phase)

- Boite de dialogue modale
- Boite de dialogue non modale
- Message d'information
- Message d'avertissement
- Message d'erreur (métier)
- Message de confirmation
- Message d'erreur (exception)

Results

Export

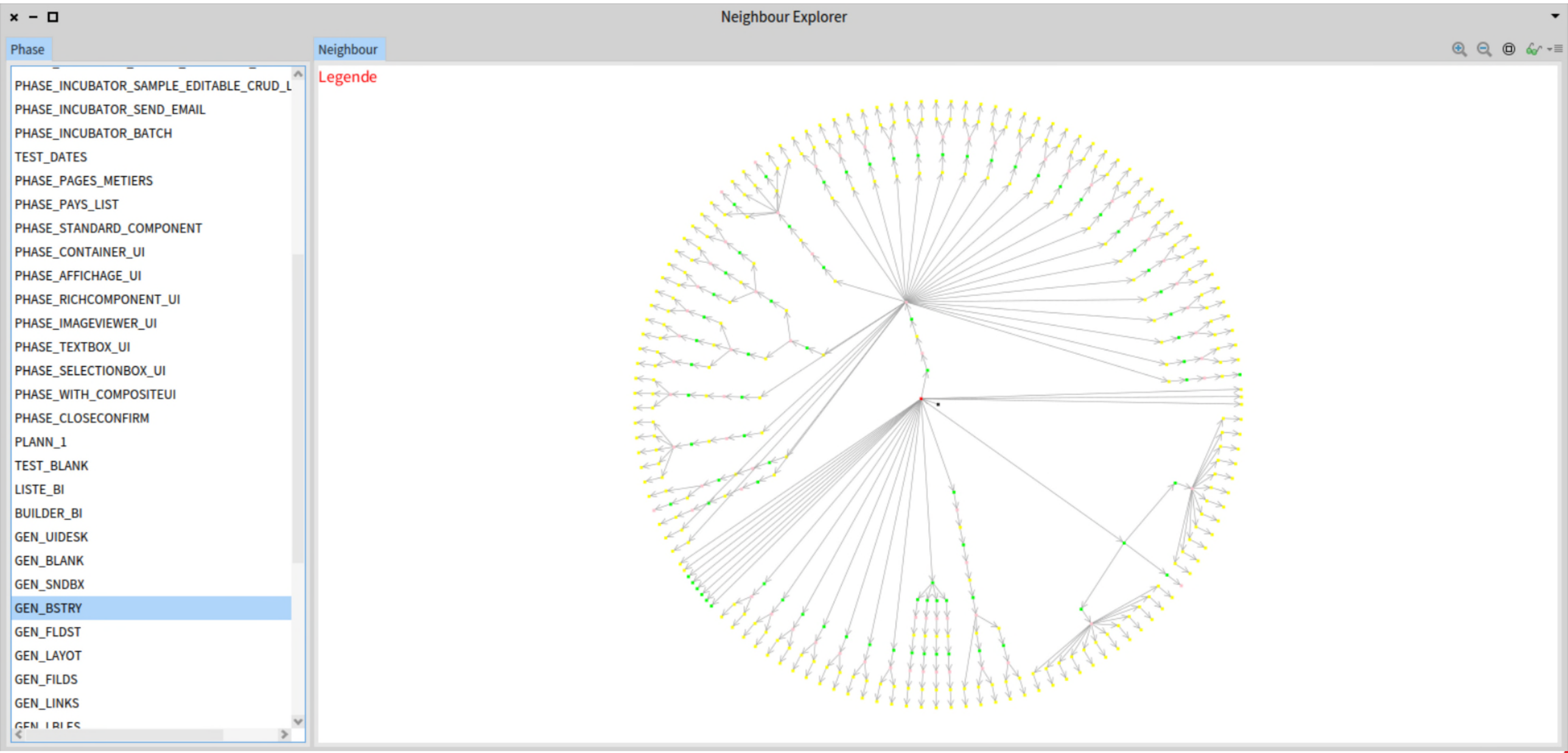
Etiquettes formattées (pour les listes)

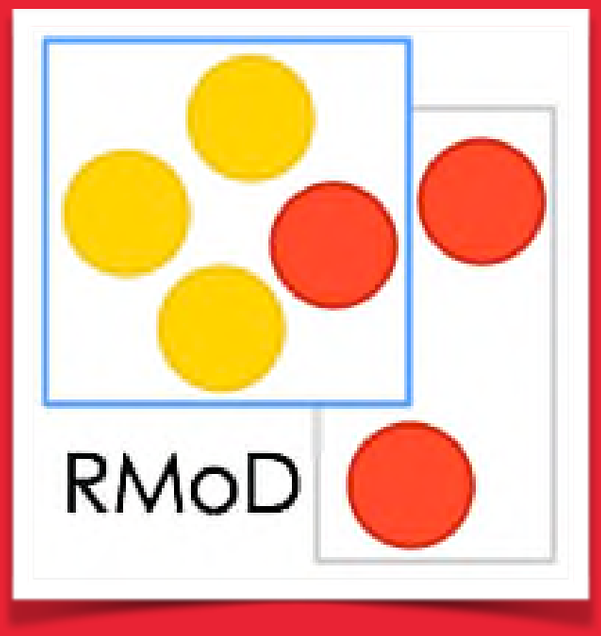
Montant :	10 000 000,00 €
Pourcentage :	1,50%
Booléen (1) :	Oui
Booléen (0) :	Non
Date :	26/06/2018
Durée :	3 ans, 2 mois et 1 jour
Enumération :	jour(s)
Entier :	999999999
Entier long :	9999999999999999
Entier short :	999

Etiquettes formattées (pour les listes)

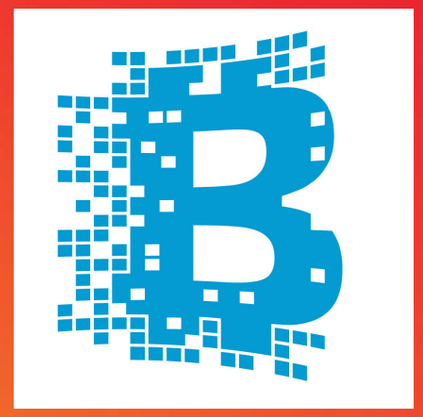
Montant :Pourcentage :Booléen (1) :Booléen (0) :Date :Durée :Enumération :Entier :Entier long :Entier short :

Tool to explore





Tools for Blockchain Software Engineering



Warning!

Research effort started two years ago

Financed by UTOCAT

SmartAnvil: First platform to build blockchain software analysis tools

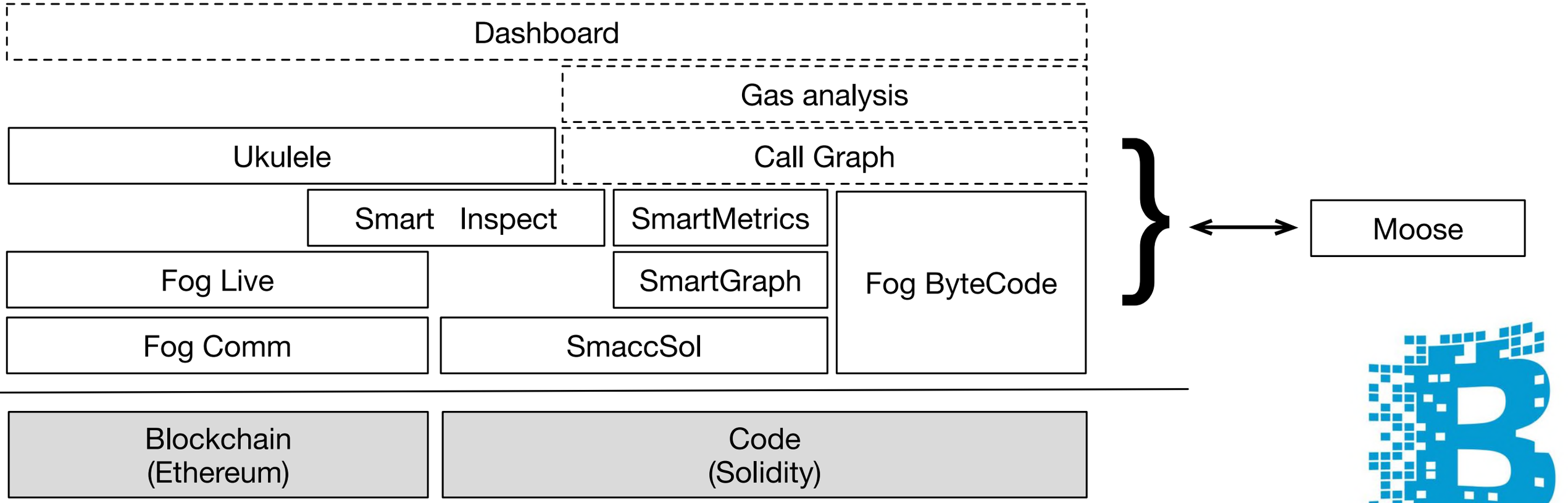
Still ongoing work on many points

Solidity Static Analysis
Ethereum Inspector
Smart Debug
Blockchain Query Language
Smart Contract Metrics
Contract Visualization

Inria

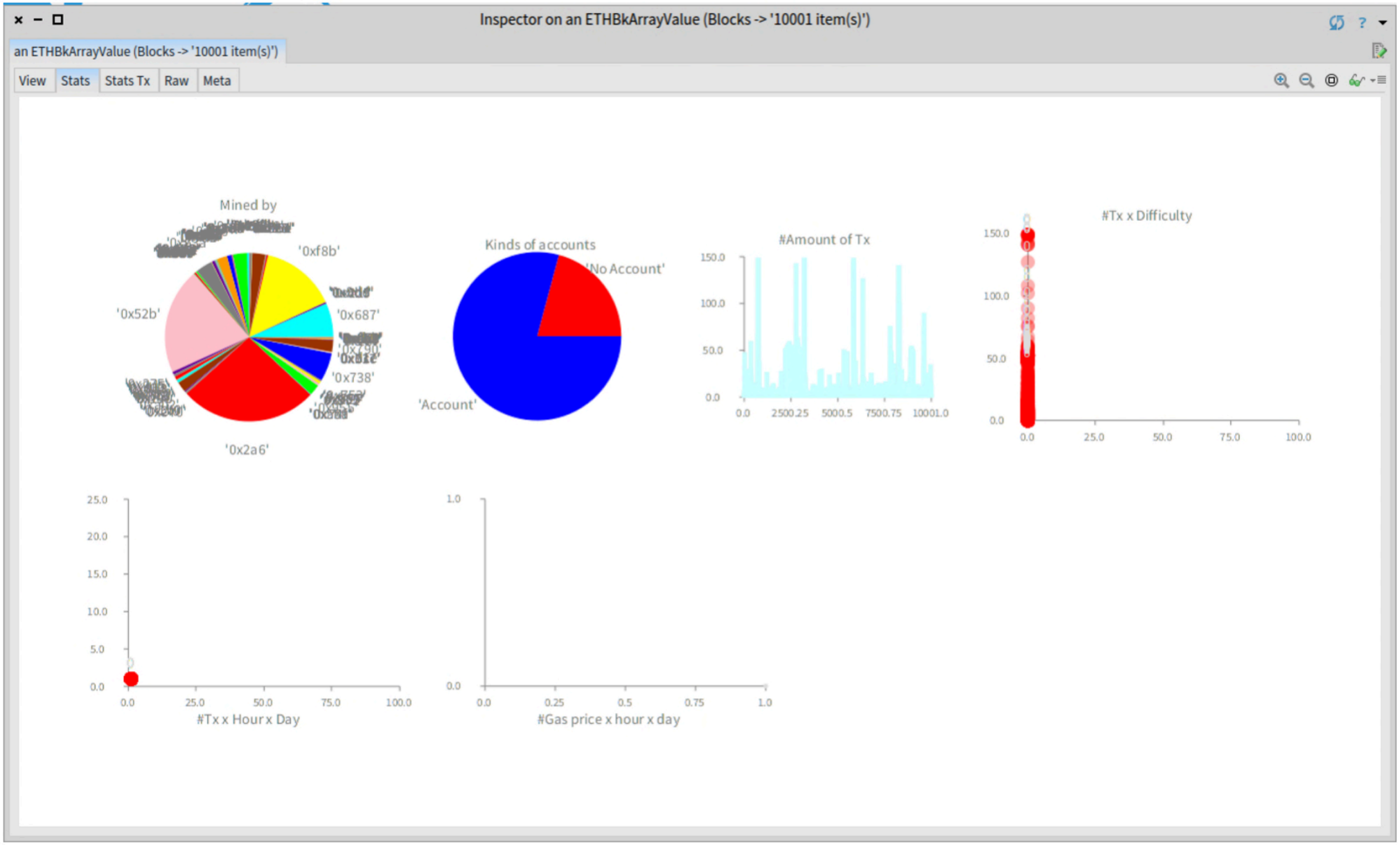
RMoD

SmartAnvil: Open-source platform for build tools for SE-Blockchain



Can we build a smart contract dashboard?

Blockchain Visualisation



How to understand a bug in deployed smart contracts?

What is the value of x in function y for contract 0xb45787ac….?

Ethereum Inspector
[SmartInspect, IWBOSE/SANER 2018]

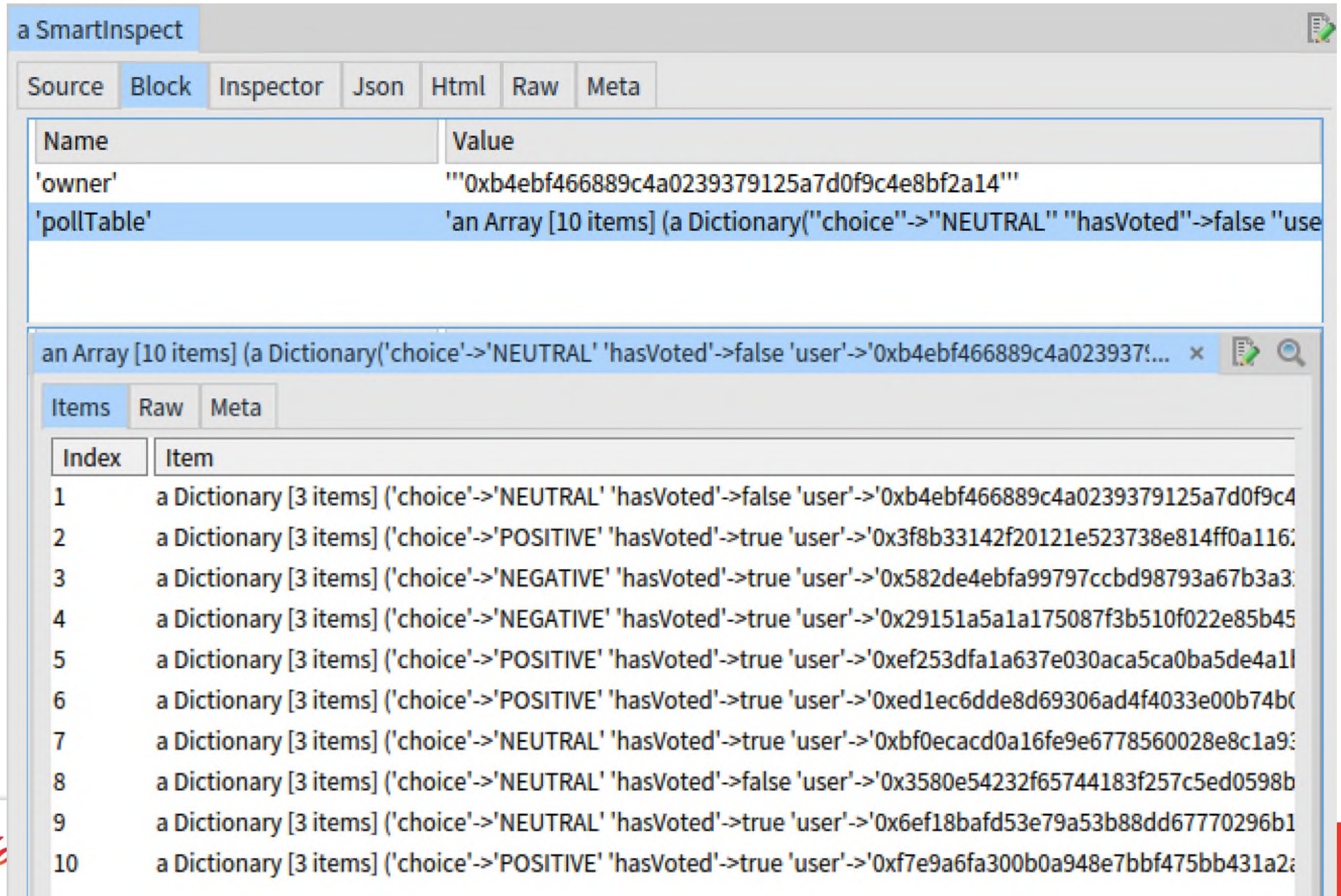
Deployed Smart Contracts are opaque

Debugging contracts very tedious and time-consuming.

No printf :)

Contract inspection can help developer & companies

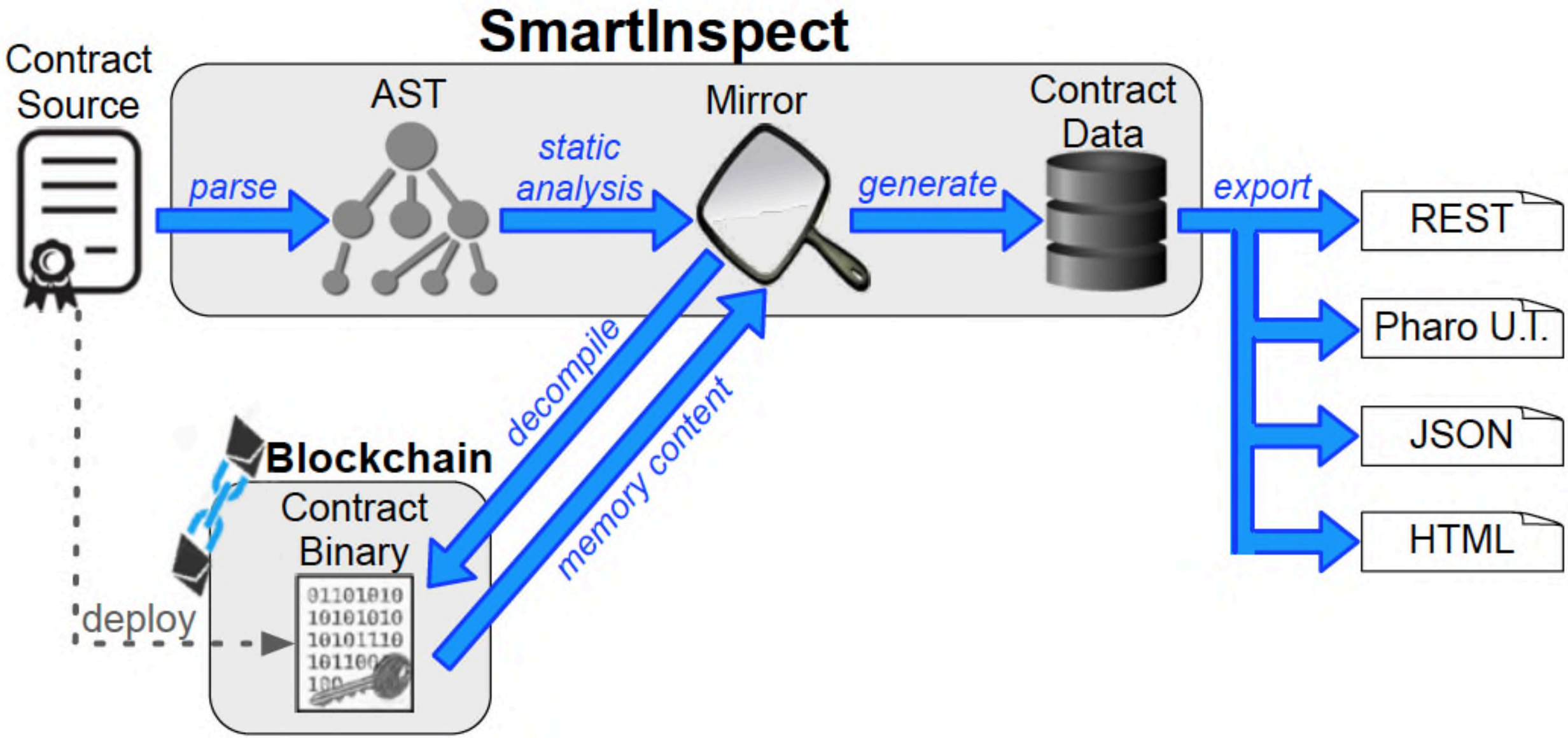
Ethereum Inspector



Ethereum Inspector

- Pluggable
- Interactive
- Consistent
- Reusable
- Full type support

Ethereum Inspector Architecture



How to debug a smart contract?

Smart Debug

Full solidity bytecode interpreter

Storage simulation

Step and memory access

Call semantics

Gas consumption

Smart-Anvil EVM Debugger

Activation | Interpreter | Solidity co

```
pragma solidity ^0.4.2;

contract Public3StatesPoll {

  /* Type Definition */
   enum Choice { POSITIVE , NEGATIVE , NEUTRAL
}
   struct Columns {  address user; Choice choice ;
bool hasVoted; }

  /* Properties */
   Columns[] pollTable;
   address owner;


   /* Constructor */
   function Public3StatesPoll () {
      owner = msg.sender;
   }
   /* Functions */
   function isRegistered (address voterAccount)
returns(bool registered){
      registered = (voterIndex (voterAccount) >= 0);
   }
   function voterIndex (address voterAccount)
returns(int256 index){
      for(uint x = 0; x < pollTable.length; x = x +1) {
        if (pollTable[x].user == voterAccount) {
          return int(x);
```

IP	#	instruction
16r0		PUSH1 0x80
16r2		PUSH1 0x40
16r4		MSTORE
16r5		CALLVALUE
16r6		DUP1
16r7		ISZERO
16r8		PUSH2 0x10
16rB		JUMPI
16rC		PUSH1 0x0
16rE		DUP1
16rF		REVERT
16r10		JUMPDEST
16r11		POP
16r12		CALLER
16r13		PUSH1 0x1
16r15		PUSH1 0x0
16r17		PUSH2 0x100
16r1A		EXP
16r1B		DUP2
16r1C		SLOAD
16r1D		DUP2
16r1E		PUSH20 0xFFFFFFFFFFFFFF
16r33		MUL
16r34		NOT
16r35		AND
16r36		SWAP1
16r37		DUP4
16r38		PUSH20 0xFFFFFFFFFFFFFF

stack

Result	instructions	#	#	#	#	#	#
[0x81BFBA830	)OR	0x81BFBA8	0x0				
[0x1][0x81BFB	/SWAP1	0x81BFBA8	0x1				
[]	SSTORE	0x1	0x81BFBA8				
[]	POP	0x81BFBA8					
[0x608]	PUSH2 0x608						
[0x608][0x608]	DUP1	0x608					
[0x61]	PUSH2 0x61						
[0x0]	PUSH1 0x0						
[]	CODECOPY	0x0	0x61	0x608			
[0x0]	PUSH1 0x0						
[]	RETURN	0x0	0x608				

Finished

Memory | Store

- Node storage - (1 entries) - ce9418d9-4b38-0d00-81ed-4653022e7a52
  - Storage (0x4164FEF5054C0EEAC71E2C692D10BEC4226AA6DD) - FogEVMBucketStorage
    - #'&contract' => a FogEVMContract
      0x1 => 0x81BFBA8301A073A010E9FB71FAB514E53D1CC3F0

Logs | Errors

Filter...

Smart-Anvil EVM Debugger

Activation | Interpreter | Solidity co

From: 0x81BFBA8301A073A010E9FB71FAB5
To:
Contract: 0x4164FEF5054C0EEAC71E2C692D10
Context: nil
Value: 0
Gas: 3000000
Gas price: 1
Input:
'0x6080604052348015610010576000 80fd5b5

Return:

Binary Ret 6080604052600436106100785760003

IP	#	instruction
	16r0	PUSH1 0x80
	16r2	PUSH1 0x40
	16r4	MSTORE
	16r5	CALLVALUE
	16r6	DUP1
	16r7	ISZERO
	16r8	PUSH2 0x10
	16rB	JUMPI
	16rC	PUSH1 0x0
	16rE	DUP1
	16rF	REVERT
	16r10	JUMPDEST
	16r11	POP
	16r12	CALLER
	16r13	PUSH1 0x1
	16r15	PUSH1 0x0
	16r17	PUSH2 0x100
	16r1A	EXP
	16r1B	DUP2
	16r1C	SLOAD
	16r1D	DUP2
	16r1E	PUSH20 0xFFFFFFFFFFFFFF
	16r33	MUL
	16r34	NOT
	16r35	AND
	16r36	SWAP1
	16r37	DUP4
	16r38	PUSH20 0xFFFFFFFFFFFFFF

stack

Result	instructions	#	#	#	#	#	#
[0x81BFBA8301	OR	0x81BFBA8	0x0				
[0x1][0x81BFB	SWAP1	0x81BFBA8	0x1				
[]	SSTORE	0x1	0x81BFBA8				
[]	POP	0x81BFBA8					
[0x608]	PUSH2 0x608						
[0x608][0x608]	DUP1	0x608					
[0x61]	PUSH2 0x61						
[0x0]	PUSH1 0x0						
[]	CODECOPY	0x0	0x61	0x608			
[0x0]	PUSH1 0x0						
[]	RETURN	0x0	0x608				

Finished

Memory | Store

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	+10	+11	+12	+13	+14	+15	+16
0	60	80	60	40	52	60	04	36	10	61	00	78	57	60	00	35	7c	01	00	00	00	00	00
20	00	00	00	00	00	00	00	00	00	00	00	00	00	00	90	04	63	ff	ff	ff	ff	16	80
40	57	80	63	b3	f9	8a	dc	14	61	00	ad	57	80	63	c3	c5	a5	47	14	61	00	dd	57
60	38	57	80	63	ce	a6	cf	5e	14	61	01	8f	57	80	63	f4	ab	9a	df	14	61	01	be
80	15	61	00	89	57	60	00	80	fd	5b	50	61	00	ab	60	04	80	36	03	81	01	90	80
A0	90	92	91	90	50	50	50	61	02	15	56	5b	00	5b	34	80	15	61	00	b9	57	60	00
C0	80	36	03	81	01	90	80	80	35	60	ff	16	90	60	20	01	90	92	91	90	50	50	50
E0	15	61	00	e9	57	60	00	80	fd	5b	50	61	01	1e	60	04	80	36	03	81	01	90	80
100	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	16	90	60	20	01	90	92	91	90

Logs | Errors

Filter...

How to identify one transaction?

How to identify a contract?

In Ethereum:

- **Hash Access memory**

  If you lose the hash of your transaction…

  you are in troubles….

- **Massive data**

  if you have to look for it one by one it would be impossible

- **Data opaqueness**

  You may not have any help from the available tools to understand what you are looking at

Blockchain Query Language

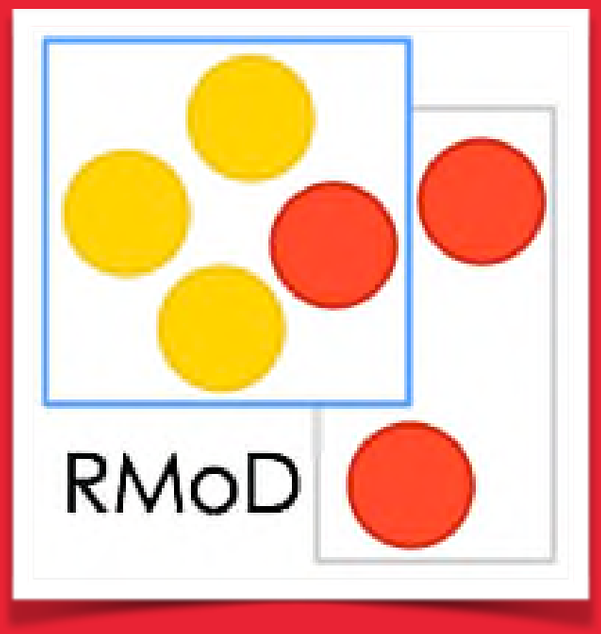
```
1 SELECT block.parent.number, block.hash,
      block.timestamp, block.number,
      block.amountOfTransactions
2 FROM ethereum.blocks AS block
3 WHERE block.timestamp BETWEEN date('2016-01-01')
      AND now() AND block.transactions.size >10
4 ORDER BY block.transactions.size
5 LIMIT 100;
```

block.parent().number	block.hash	block.timestamp	block.number	block.amou
779523	8969218	2016-01-01T05:53:07+01:00	779524	11
779112	104199555	2016-01-01T03:51:04+01:00	779113	11
779505	226022799	2016-01-01T05:48:01+01:00	779506	11
782753	96179627	2016-01-01T21:48:43+01:00	782754	11
780348	239565626	2016-01-01T10:10:40+01:00	780349	11
779113	101468812	2016-01-01T03:51:20+01:00	779114	11
782218	39336575	2016-01-01T19:18:16+01:00	782219	11
778657	129555478	2016-01-01T01:48:18+01:00	778658	11
779281	249003683	2016-01-01T04:38:40+01:00	779282	11
780351	103525535	2016-01-01T10:11:05+01:00	780352	11
781319	259274786	2016-01-01T14:56:03+01:00	781320	11
779829	186089684	2016-01-01T07:21:49+01:00	779830	12

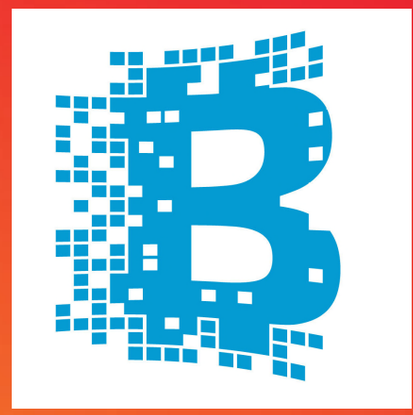
Blockchain Query Language

Items | Raw | Meta

Index	Item
1	an OrderedCollection [5 items] ('block..parent.()'->'Block- 779523 - 4 transaction(s) ' 'block.hash.'->8969218 'block.timestamp.'->2016-01-01T05:53:07+01:00 'block.number.'->779524 'block..amountOfTransactions.()'->11)
2	an OrderedCollection [5 items] ('block..parent.()'->'Block- 779112 - 1 transaction(s) ' 'block.hash.'->104199555 'block.timestamp.'->2016-01-01T03:51:04+01:00 'block.number.'->779113 'block..amountOfTransactions.()'->11)
3	an OrderedCollection [5 items] ('block..parent.()'->'Block- 779505 - 0 transaction(s) ' 'block.hash.'->226022799 'block.timestamp.'->2016-01-01T05:48:01+01:00 'block.number.'->779506 'block..amountOfTransactions.()'->11)
4	an OrderedCollection [5 items] ('block..parent.()'->'Block- 782753 - 0 transaction(s) ' 'block.hash.'->96179627 'block.timestamp.'->2016-01-01T21:48:43+01:00 'block.number.'->782754 'block..amountOfTransactions.()'->11)
5	an OrderedCollection [5 items] ('block..parent.()'->'Block- 780348 - 8 transaction(s) ' 'block.hash.'->239565626 'block.timestamp.'->2016-01-01T10:10:40+01:00 'block.number.'->780349 'block..amountOfTransactions.()'->11)
6	an OrderedCollection [5 items] ('block..parent.()'->'Block- 779113 - 11 transaction(s) ' 'block.hash.'->101468812 'block.timestamp.'->2016-01-01T03:51:20+01:00 'block.number.'->779114 'block..amountOfTransactions.()'->11)
7	an OrderedCollection [5 items] ('block..parent.()'->'Block- 782218 - 0 transaction(s) ' 'block.hash.'->39336575 'block.timestamp.'->2016-01-01T19:18:16+01:00 'block.number.'->782219 'block..amountOfTransactions.()'->11)
8	an OrderedCollection [5 items] ('block..parent.()'->'Block- 778657 - 2 transaction(s) ' 'block.hash.'->129555478 'block.timestamp.'->2016-01-01T01:48:18+01:00 'block.number.'->778658 'block..amountOfTransactions.()'->11)
9	an OrderedCollection [5 items] ('block..parent.()'->'Block- 779281 - 3 transaction(s) ' 'block.hash.'->249003683 'block.timestamp.'->2016-01-01T04:38:40+01:00 'block.number.'->779282 'block..amountOfTransactions.()'->11)
10	an OrderedCollection [5 items] ('block..parent.()'->'Block- 780351 - 8 transaction(s) ' 'block.hash.'->103525535 'block.timestamp.'->2016-01-01T10:11:05+01:00 'block.number.'->780352 'block..amountOfTransactions.()'->11)
11	an OrderedCollection [5 items] ('block..parent.()'->'Block- 781319 - 0 transaction(s) ' 'block.hash.'->259274786 'block.timestamp.'->2016-01-01T14:56:03+01:00 'block.number.'->781320 'block..amountOfTransactions.()'->11)
12	an OrderedCollection [5 items] ('block..parent.()'->'Block- 779829 - 1 transaction(s) ' 'block.hash.'->186089684 'block.timestamp.'->2016-01-01T07:21:49+01:00 'block.number.'->779830 'block..amountOfTransactions.()'->12)
13	an OrderedCollection [5 items] ('block..parent.()'->'Block- 781563 - 1 transaction(s) ' 'block.hash.'->192796997 'block.timestamp.'->2016-01-01T16:11:22+01:00 'block.number.'->781564 'block..amountOfTransactions.()'->12)
14	an OrderedCollection [5 items] ('block..parent.()'->'Block- 778286 - 0 transaction(s) ' 'block.hash.'->148395039 'block.timestamp.'->2016-01-01T00:05:16+01:00 'block.number.'->778287 'block..amountOfTransactions.()'->12)
15	an OrderedCollection [5 items] ('block..parent.()'->'Block- 780353 - 3 transaction(s) ' 'block.hash.'->245360810 'block.timestamp.'->2016-01-01T10:11:23+01:00 'block.number.'->780354 'block..amountOfTransactions.()'->12)
16	an OrderedCollection [5 items] ('block..parent.()'->'Block- 781498 - 1 transaction(s) ' 'block.hash.'->167190109 'block.timestamp.'->2016-01-01T15:52:34+01:00 'block.number.'->781499 'block..amountOfTransactions.()'->12)
17	an OrderedCollection [5 items] ('block..parent.()'->'Block- 780917 - 3 transaction(s) ' 'block.hash.'->180827207 'block.timestamp.'->2016-01-01T13:04:17+01:00 'block.number.'->780918 'block..amountOfTransactions.()'->12)
18	an OrderedCollection [5 items] ('block..parent.()'->'Block- 781459 - 1 transaction(s) ' 'block.hash.'->255862972 'block.timestamp.'->2016-01-01T15:40:24+01:00 'block.number.'->781460 'block..amountOfTransactions.()'->12)
19	an OrderedCollection [5 items] ('block..parent.()'->'Block- 780687 - 5 transaction(s) ' 'block.hash.'->118747668 'block.timestamp.'->2016-01-01T11:48:46+01:00 'block.number.'->780688 'block..amountOfTransactions.()'->12)
20	an OrderedCollection [5 items] ('block..parent.()'->'Block- 778514 - 0 transaction(s) ' 'block.hash.'->20851823 'block.timestamp.'->2016-01-01T01:07:21+01:00 'block.number.'->778515 'block..amountOfTransactions.()'->13)

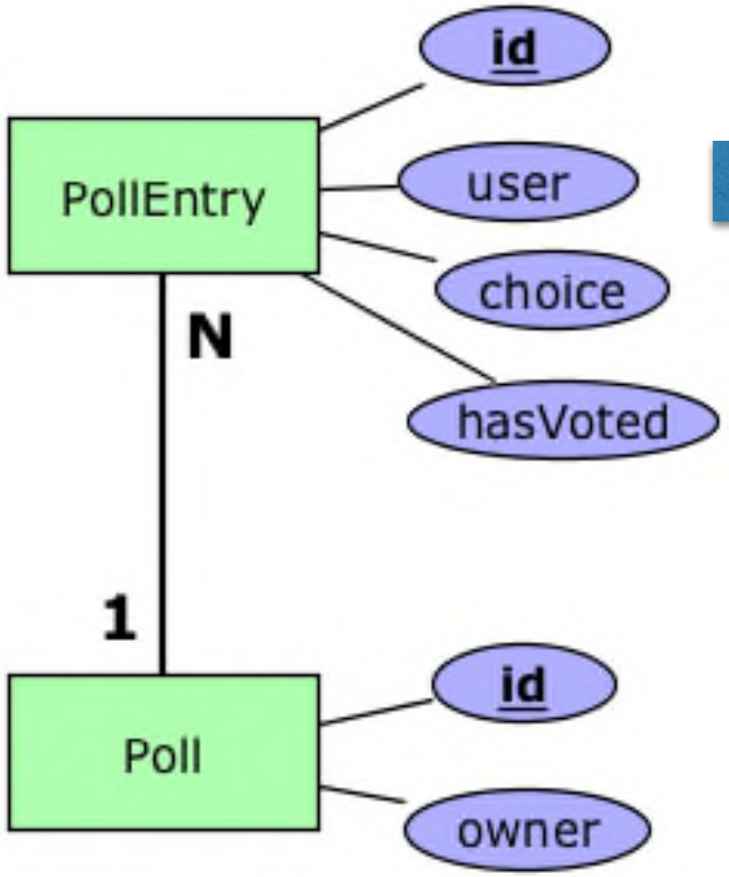


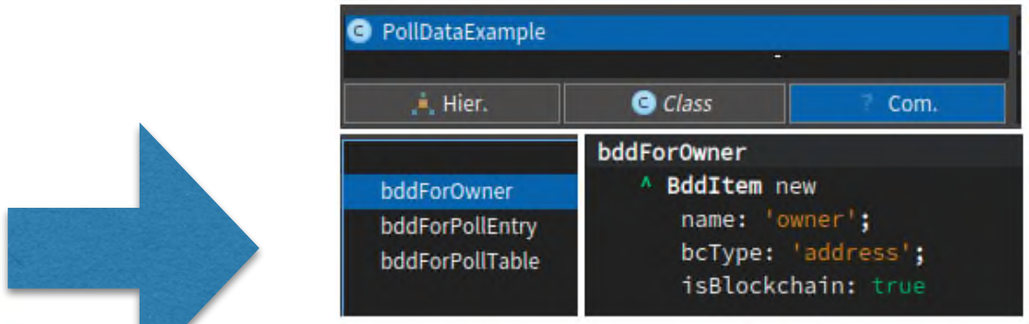
Ongoing work



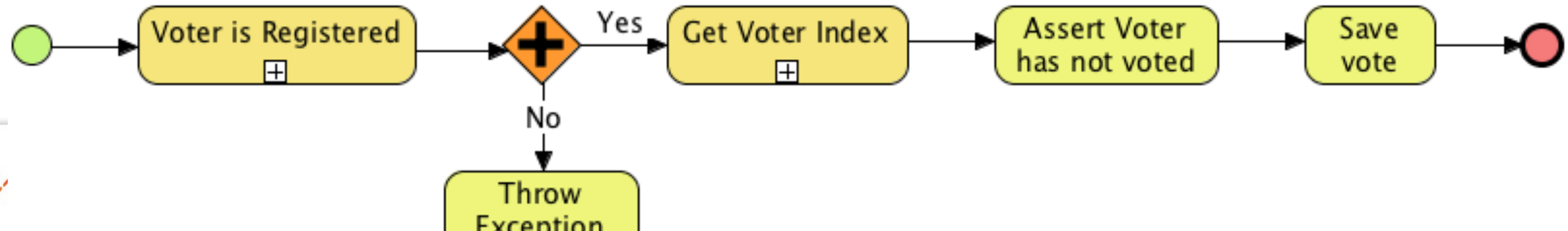
What tools need business BC engineers?

First prototypes






```
1  pragma solidity^0.4.16;
2
3  contract Poll {
4      struct PollEntry {
5          bool hasVoted;
6          Choice choice;
7          address user;
8      }
9
10     address owner;
11     PollEntry[] pollTable;
12     enum Choice {POSITIVE, NEGATIVE, NEUTRAL }
13
14 }
```



Blockchain modelisation [PhD - S. Bragagnolo - Berger-Levrault]

How to capture and model existing process?

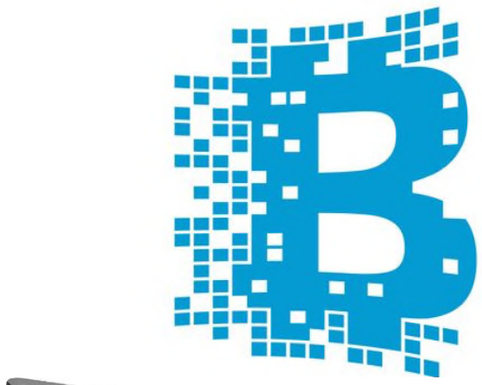
How to model and simulate domain of trust and possible architecture?

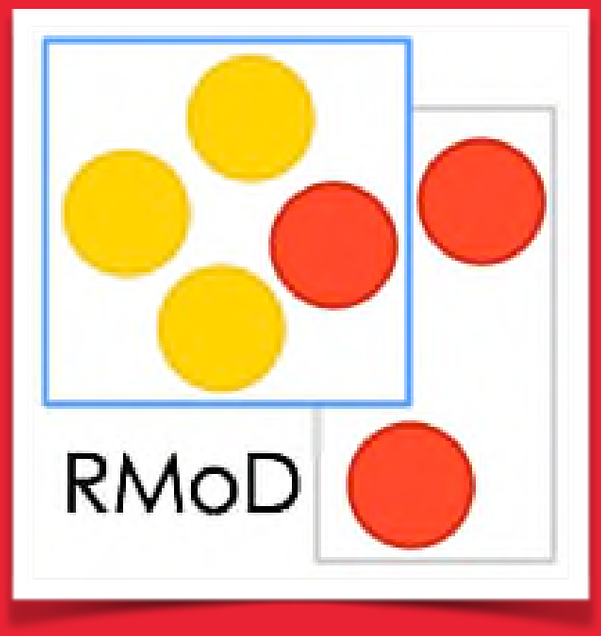
What is the impact of trust model on application architecture?

What are software metrics for Smart contracts?

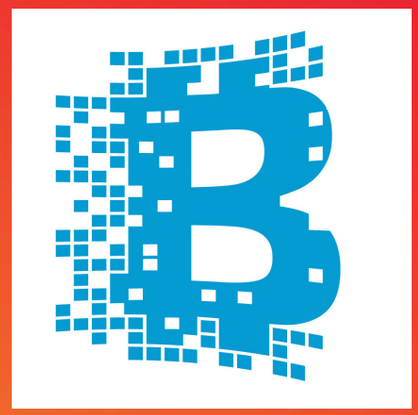
What is the quality of a Smart contract?

How to capture potential problems?





Ongoing challenges

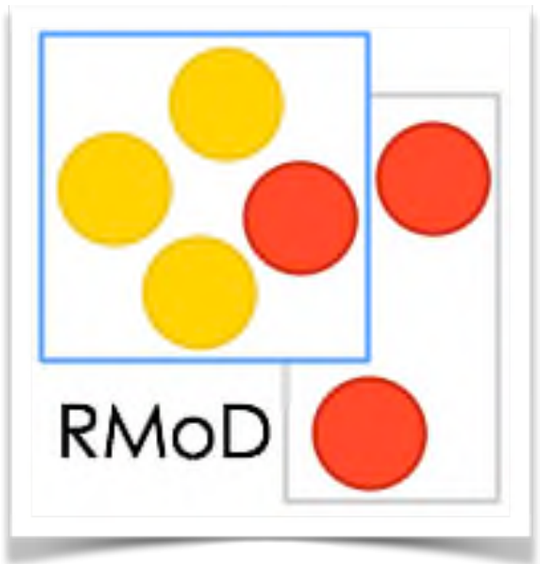


Existing applications?

How to support integration of existing applications and blockchain solution?

We believe that we can build tools for the identification of injection points.

If you are interested discuss with us.



Creating tools to tame software

Interested by your challenges

- migration help
- software assessment
- software map
- rearchitecturing
- service/micro service identification
- blockchainisation :)