2014 CEE-SEC(R)

Software Engineering Conference in Russia

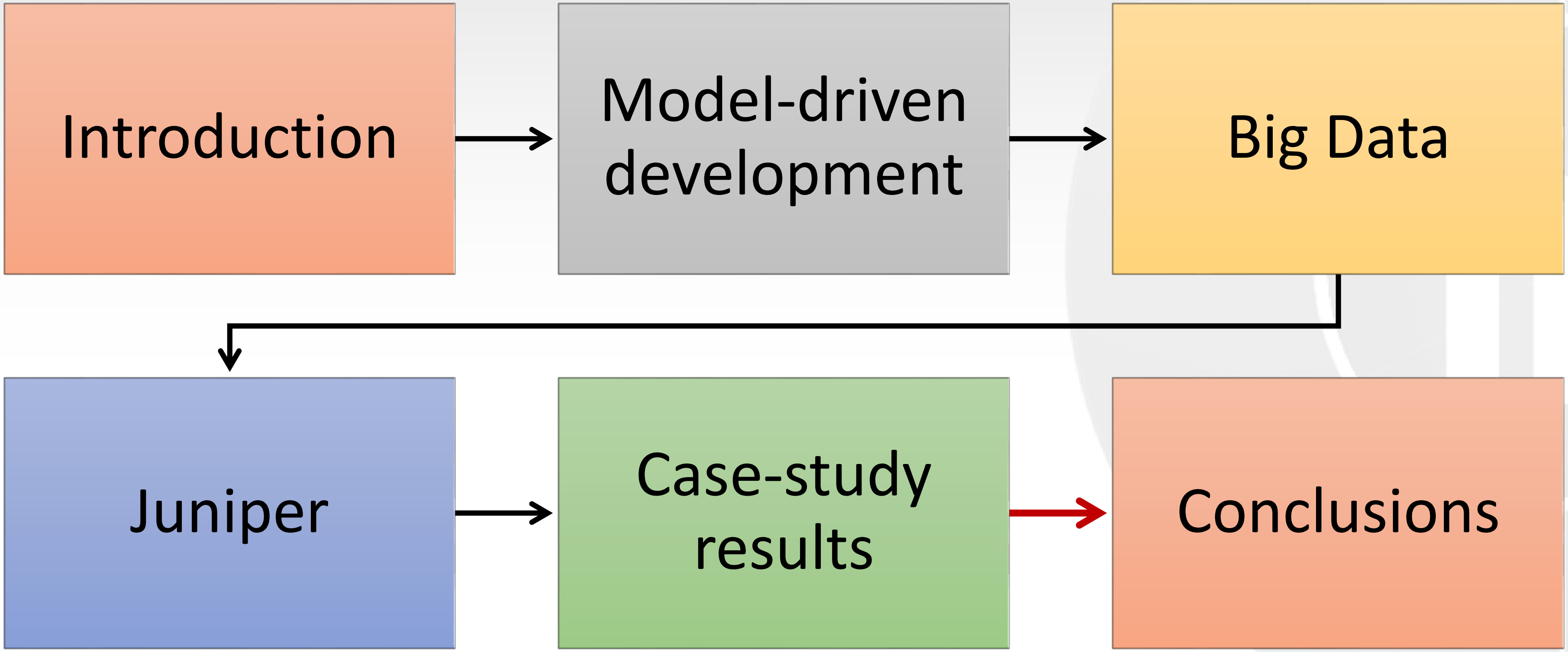
JUNIPER

JAVA PLATFORM FOR HIGH-PERFORMANCE AND REAL-TIME LARGE SCALE DATA

Towards Modeling Approach Enabling Efficient Platform for Heterogeneous Big Data Analysis

Andrey.Sadovykh@softeam.fr

Outlines



Introduction

Model-driven development

Big Data

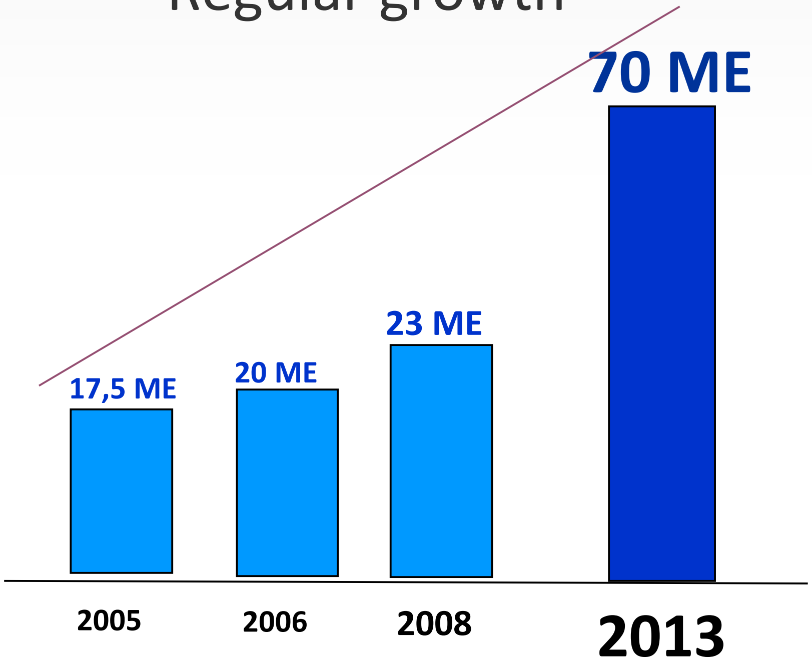
Juniper

Case-study results

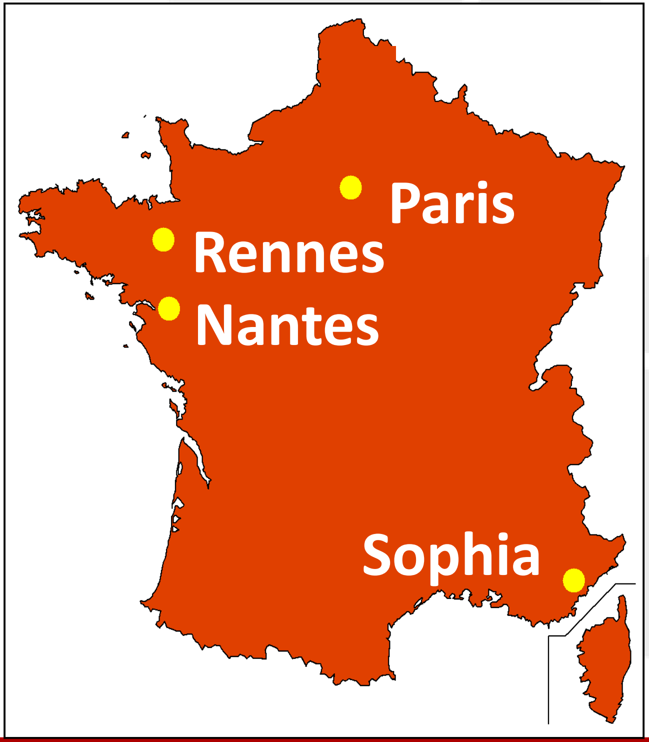
Conclusions

SOFTEAM – a French IT services / Software vendor

- SOFTEAM, a growing company
  - 25 years' experience
  - 850 experts
  - Regular growth

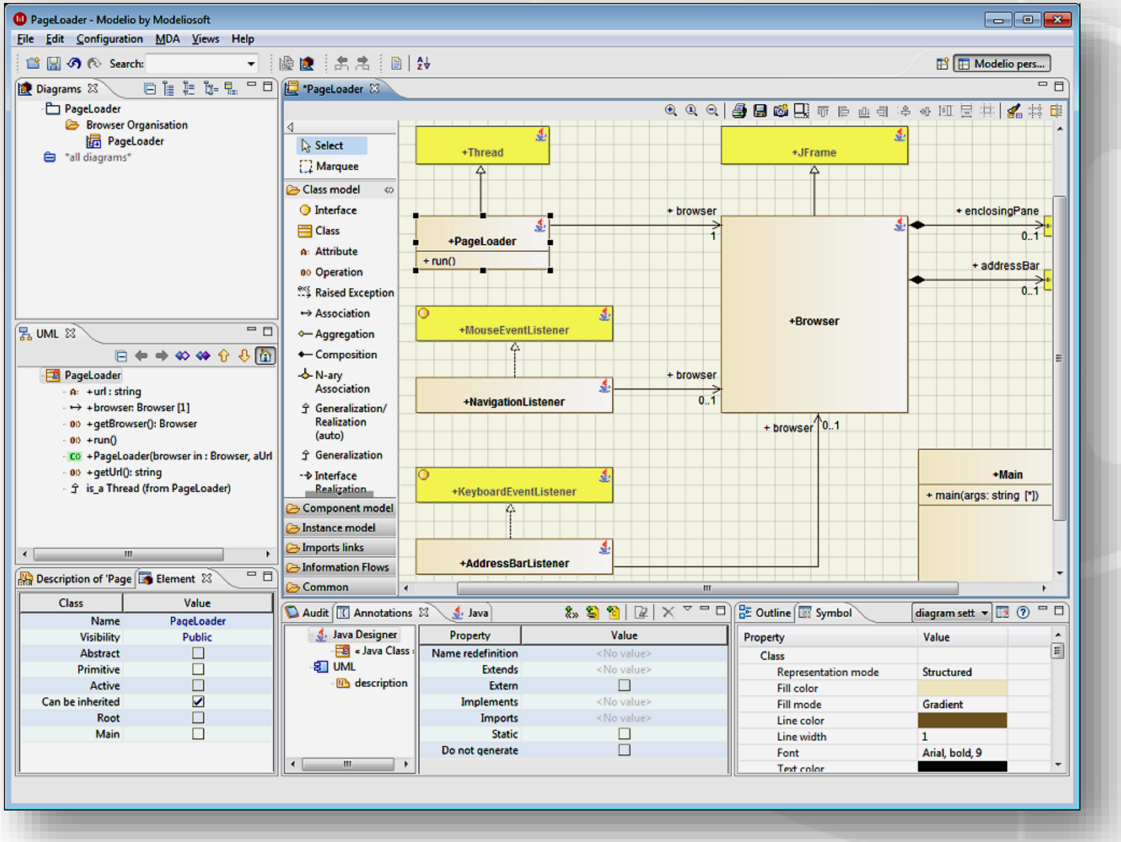


- Specialist in OO technologies, new architectures, methodologies
- Banking, Defense, Telecom, …



Modelio for Software and System Engineering

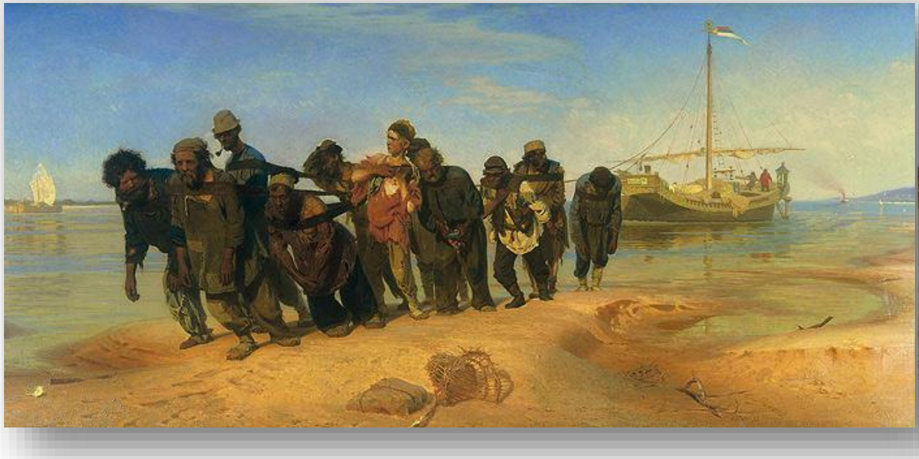
- UML editor with 20 years' history
  - CloudML
  - SysML
  - MARTE
  - Code generation
  - Documentation
  - Teamwork
- Available under open source at *Modelio.org*

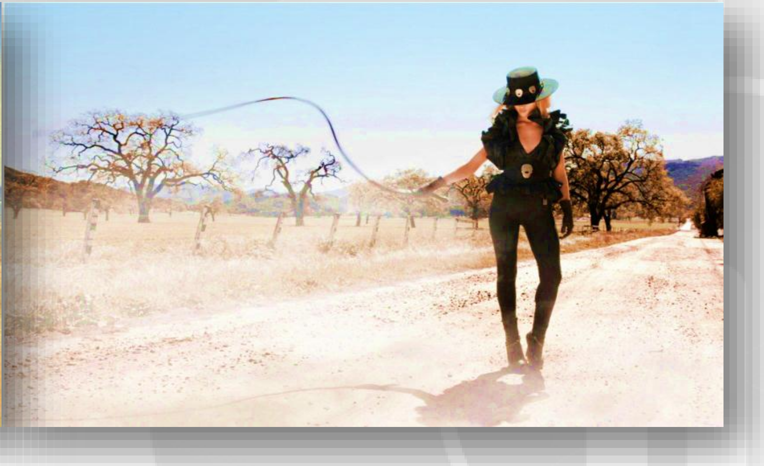


Class	Value
Name	PageLoader
Visibility	Public
Abstract	□
Primitive	□
Active	□
Can be inherited	☑
Root	□
Main	□

Property	Value
Name redefinition	<No value>
Extends	<No value>
Extern	□
Implements	<No value>
Imports	<No value>
Static	□
Do not generate	□

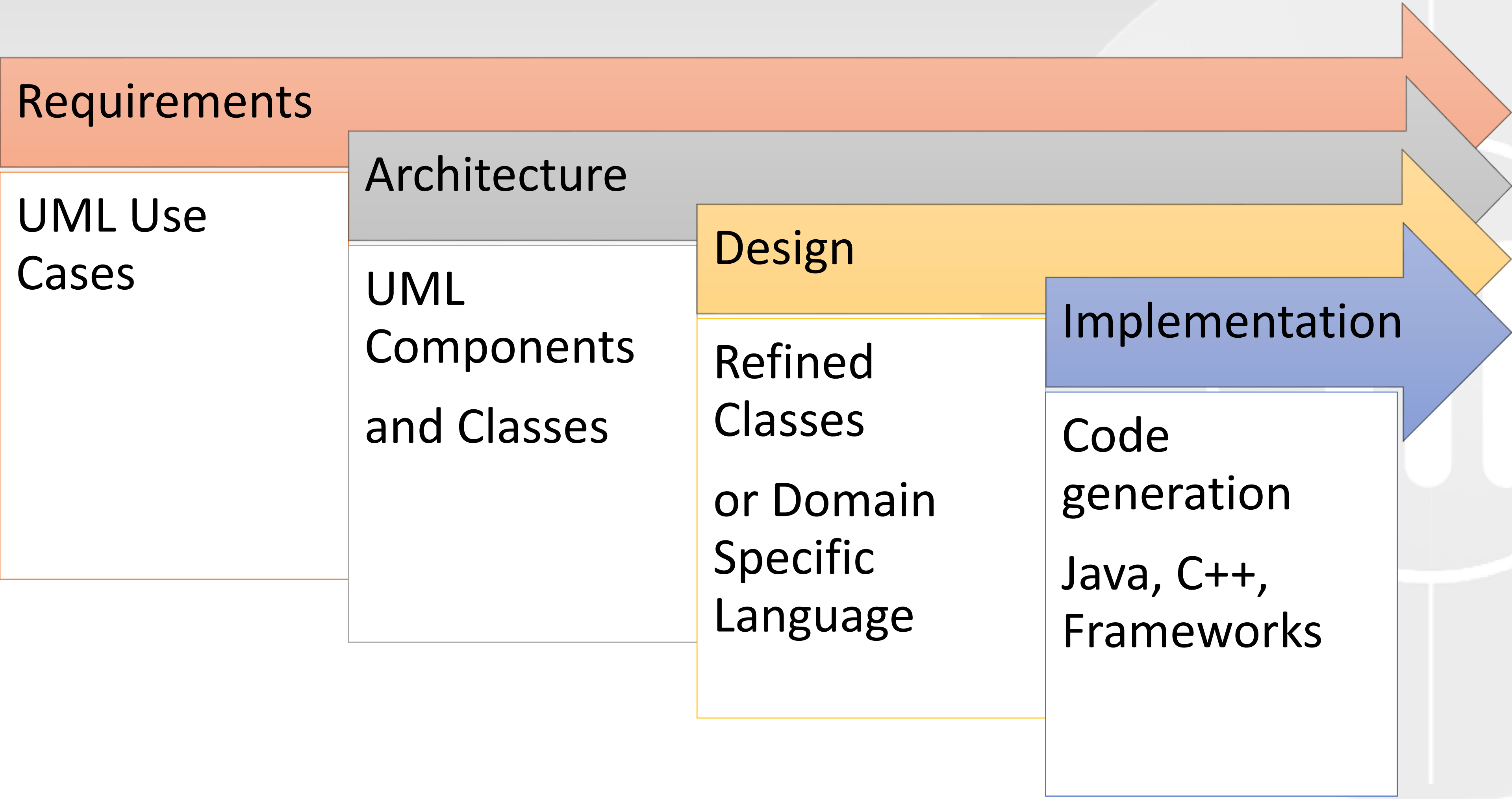
Property	Value
Class	
Representation mode	Structured
Fill color	▨
Fill mode	Gradient
Line color	■
Line width	1
Font	Arial, bold, 9
Text color	■





MODEL-DRIVEN DEVELOPMENT

It is all about models … Starting with UML



Requirements

UML Use Cases

Architecture

UML Components

and Classes

Design

Refined Classes

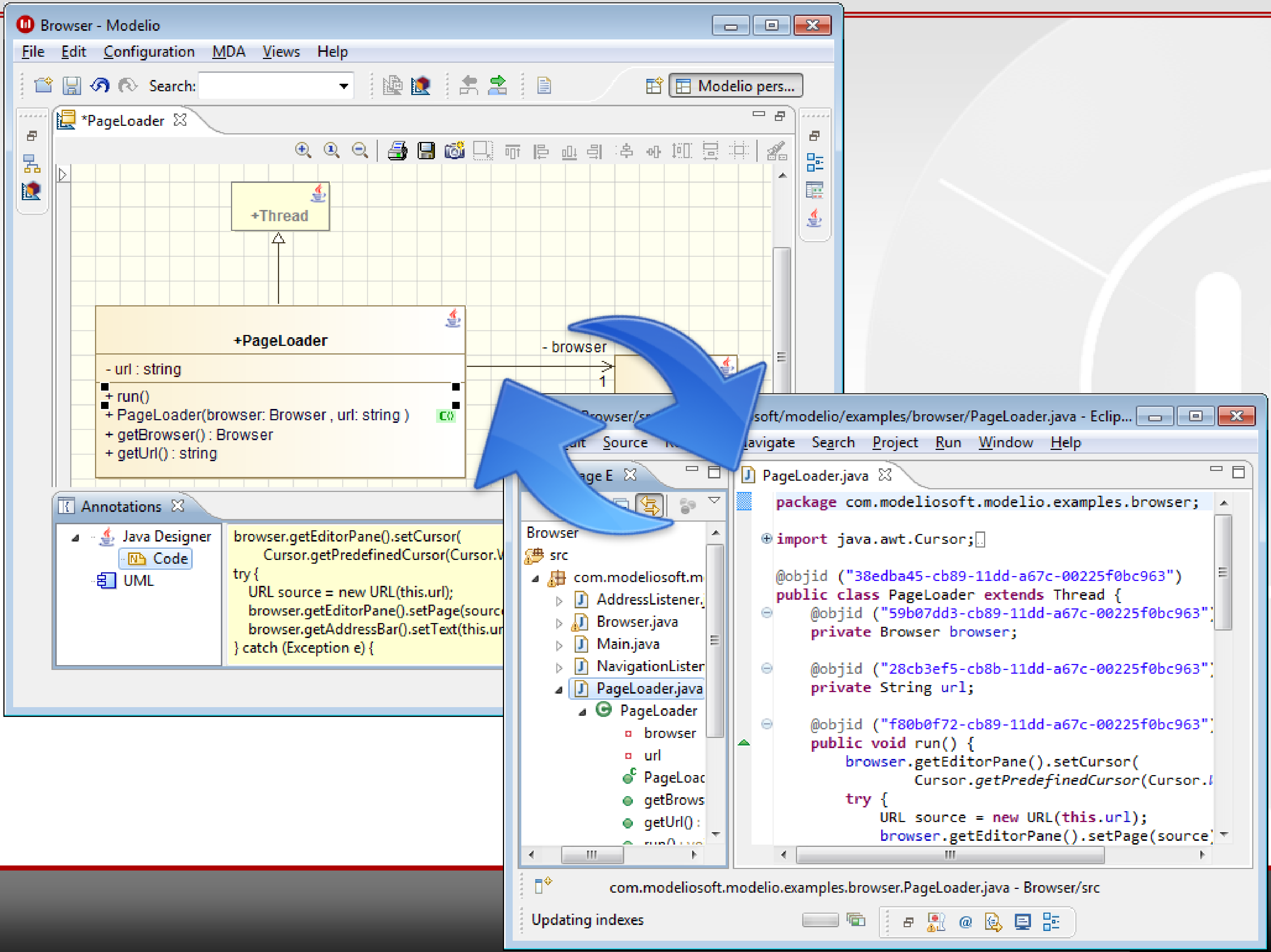
or Domain Specific Language

Implementation

Code generation

Java, C++, Frameworks

Model = Code



```
browser.getEditorPane().setCursor(
      Cursor.getPredefinedCursor(Cursor.W
try {
   URL source = new URL(this.url);
   browser.getEditorPane().setPage(sourc
   browser.getAddressBar().setText(this.ur
} catch (Exception e) {
```

```
package com.modeliosoft.modelio.examples.browser;

import java.awt.Cursor;

@objid ("38edba45-cb89-11dd-a67c-00225f0bc963")
public class PageLoader extends Thread {
    @objid ("59b07dd3-cb89-11dd-a67c-00225f0bc963"
    private Browser browser;

    @objid ("28cb3ef5-cb8b-11dd-a67c-00225f0bc963"
    private String url;

    @objid ("f80b0f72-cb89-11dd-a67c-00225f0bc963"
    public void run() {
        browser.getEditorPane().setCursor(
                Cursor.getPredefinedCursor(Cursor.
        try {
            URL source = new URL(this.url);
            browser.getEditorPane().setPage(source
```

Typical example: Control system for a frigate

- 800+ components
- Developed by 100+ engineers
- 1M+ LOC

- **MDD fosters Productivity and Quality with**
  - Code generation
  - Components reuse
  - Tracing
  - Automation

Curious DSL example: Ruby on Rails

HAML

Haml	HTML
%br{:clear => left’}	<br clear=”left”/>
%p.foo Hello	<p class=”foo”>Hello</p>
%p#foo Hello	<p id=”foo”>Hello</p>
.foo	<div class=”foo”>...</div>
#foo.bar	<div id=”foo” class=”bar”>...</div>

Cucumber and Capybara

```
Feature: User can manually add movie

Scenario: Add a movie
  Given I am on the RottenPotatoes home page
  When I follow "Add new movie"
  Then I should be on the Create New Movie page
  When I fill in "Title" with "Men In Black"
    And I should see "Men In Black"
```

What do we get from MDD?

Pros

- Design **once**, deploy **everywhere**!
- Write your transformation **once**, transform **anything**!

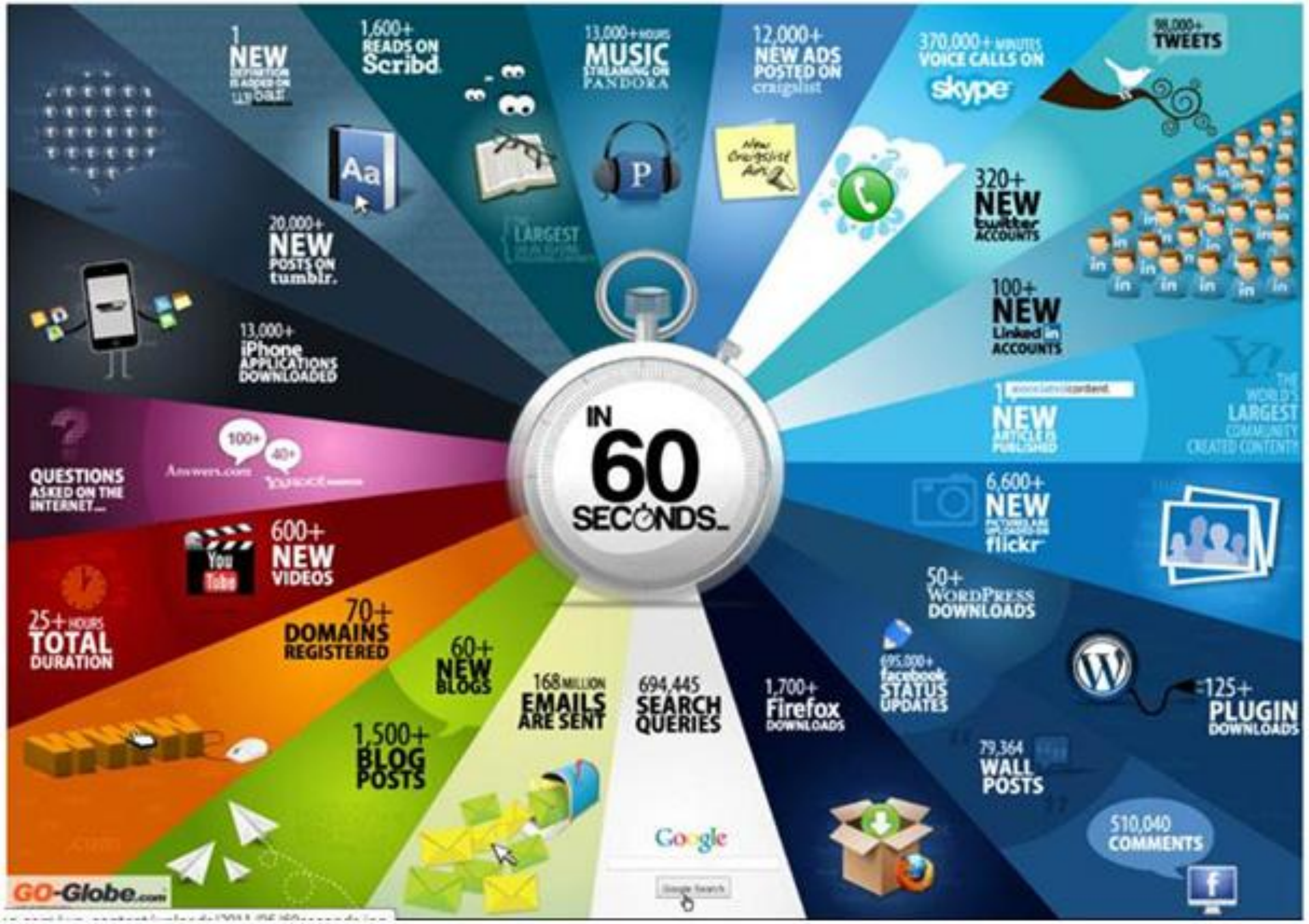
Cons

- Transformations are **hard** to write…
- How to **make sure** they are CORRECT? i.e.
  - Is there any data/semantic loss?

BIG DATA

LOOKS CLOUDY - BY FGURNEE
So You're not just a buzzword...
BIG DATA
visit www.lookscloudy.com
WWW.TOONDOO.COM

Volume, variety, velocity


IN 60 SECONDS
1 NEW DEFINITION IS ADDED ON URBAN DICTIONARY
1,600+ READS ON Scribd
13,000+ HOURS MUSIC STREAMING ON PANDORA
12,000+ NEW ADS POSTED ON craigslist
New Craigslist Ad
370,000+ MINUTES VOICE CALLS ON skype
98,000+ TWEETS
320+ NEW twitter ACCOUNTS
100+ NEW LinkedIn ACCOUNTS
1 NEW ARTICLE IS PUBLISHED
THE WORLD'S LARGEST COMMUNITY CREATED CONTENT
6,600+ NEW PICTURES ARE UPLOADED ON flickr
50+ WORDPRESS DOWNLOADS
125+ PLUGIN DOWNLOADS
695,000+ facebook STATUS UPDATES
79,364 WALL POSTS
510,040 COMMENTS
1,700+ Firefox DOWNLOADS
694,445 SEARCH QUERIES
Google
Google Search
168 MILLION EMAILS ARE SENT
1,500+ BLOG POSTS
60+ NEW BLOGS
70+ DOMAINS REGISTERED
600+ NEW VIDEOS
You Tube
25+ HOURS TOTAL DURATION
QUESTIONS ASKED ON THE INTERNET...
100+ Answers.com
40+ YAHOO! ANSWERS
13,000+ iPhone APPLICATIONS DOWNLOADED
20,000+ NEW POSTS ON tumblr.
LARGEST
GO-Globe.com

1. @-mails sent every second : **2,9 million**
2. Video uploaded to YouTube every minute: **25 hours**
3. Data processed by Google every day: **24 petabytes**
4. Tweets per day: **50 million**
5. Products ordered on Amazon per second: **73 items**

Only 0,5 % of data is analyzed

- In **2012**, 2 837EB generated - just 0,5% actually analyzed.

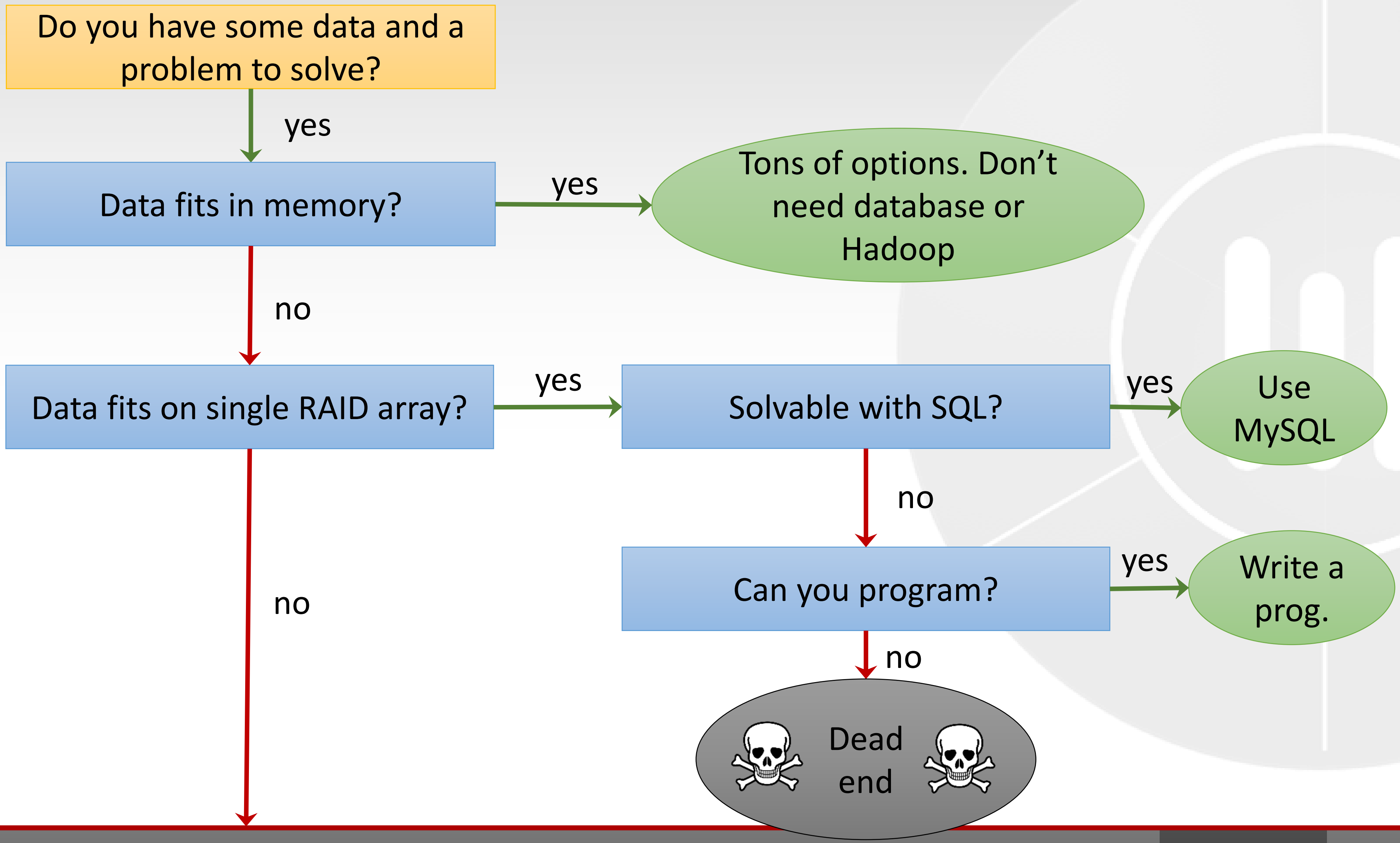
  That still amounts to 14EB (or 14.185 million terabytes)

Source: IDC & EMC

HOW'S THE BIG DATA PROJECT COMING ALONG, HOSKINS?
© D.Fletcher for CloudTweaks.com
1 0 0 1 0 1 0 1 0 1 1 1 1

SQL or Hadoop

*Inspired by: Aaron Cordova



Do you have some data and a problem to solve?

yes

Data fits in memory?

yes

Tons of options. Don't need database or Hadoop

no

Data fits on single RAID array?

yes

Solvable with SQL?

yes

Use MySQL

no

Can you program?

yes

Write a prog.

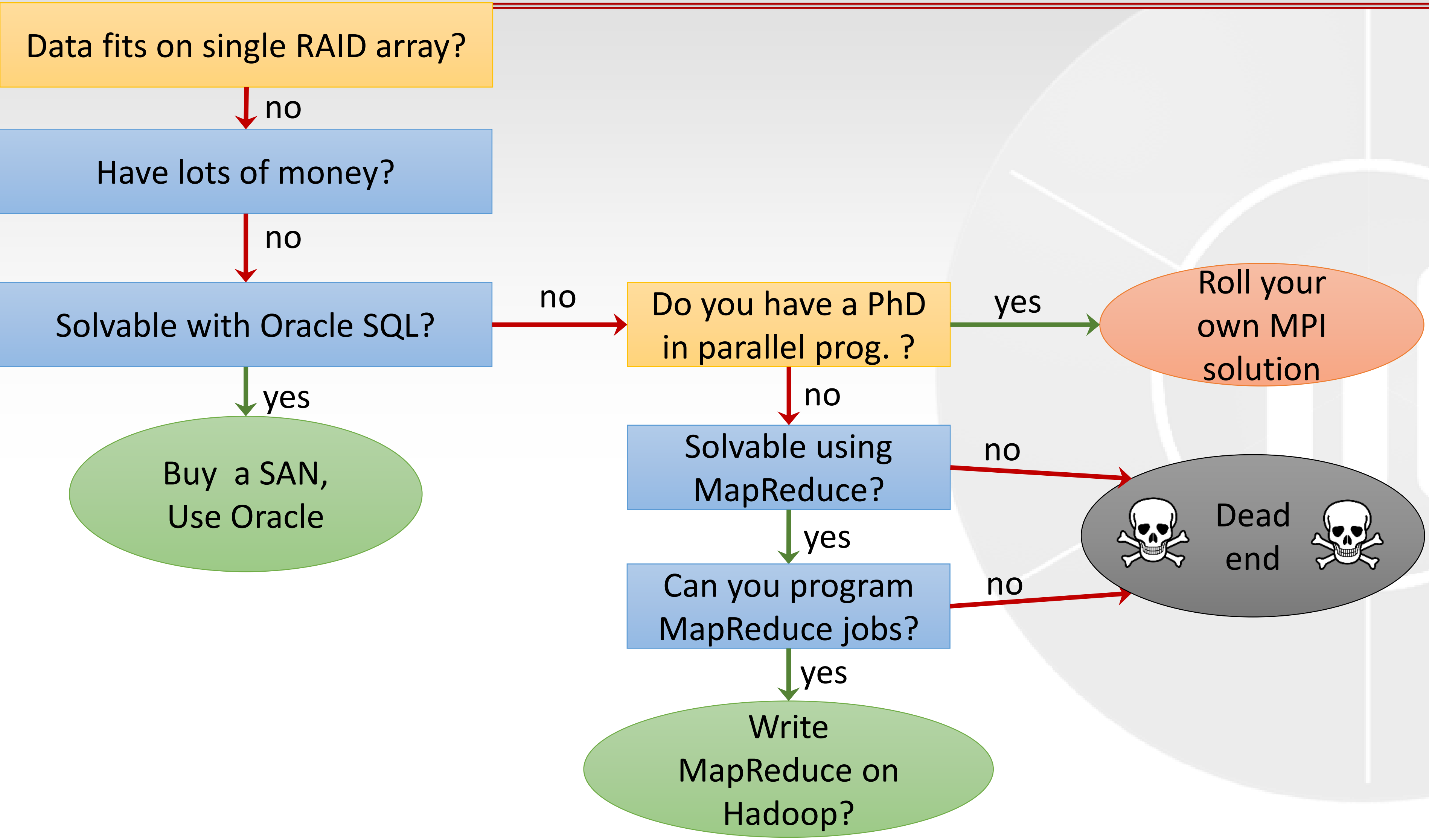
no

☠ Dead end ☠

no

SQL or Hadoop (continued)

* Inspired by: Aaron Cordova



Data fits on single RAID array?

no

Have lots of money?

no

Solvable with Oracle SQL?

yes

Buy a SAN, Use Oracle

no

Do you have a PhD in parallel prog. ?

yes

Roll your own MPI solution

no

Solvable using MapReduce?

no

Dead end

yes

Can you program MapReduce jobs?

no

yes

Write MapReduce on Hadoop?

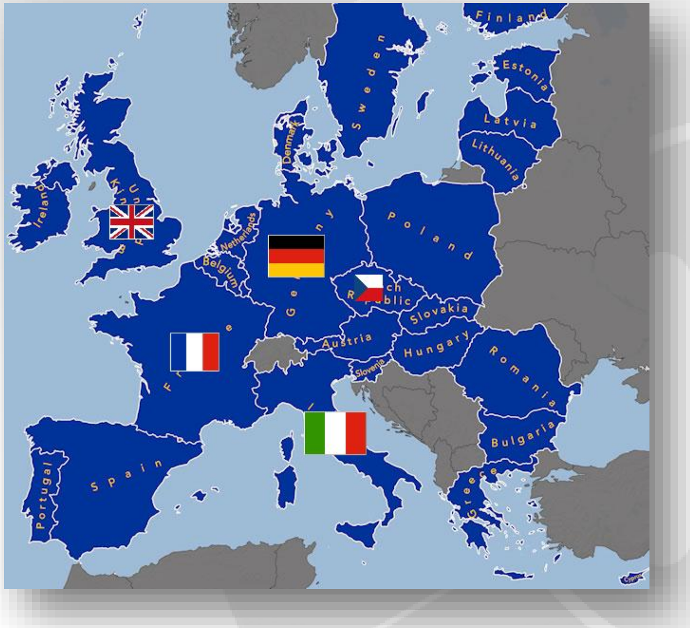
Challenges

Hadoop MapReduce is the major trend

Success relies on personnel programming skills

Many problems are not solvable with Hadoop. Real-time?

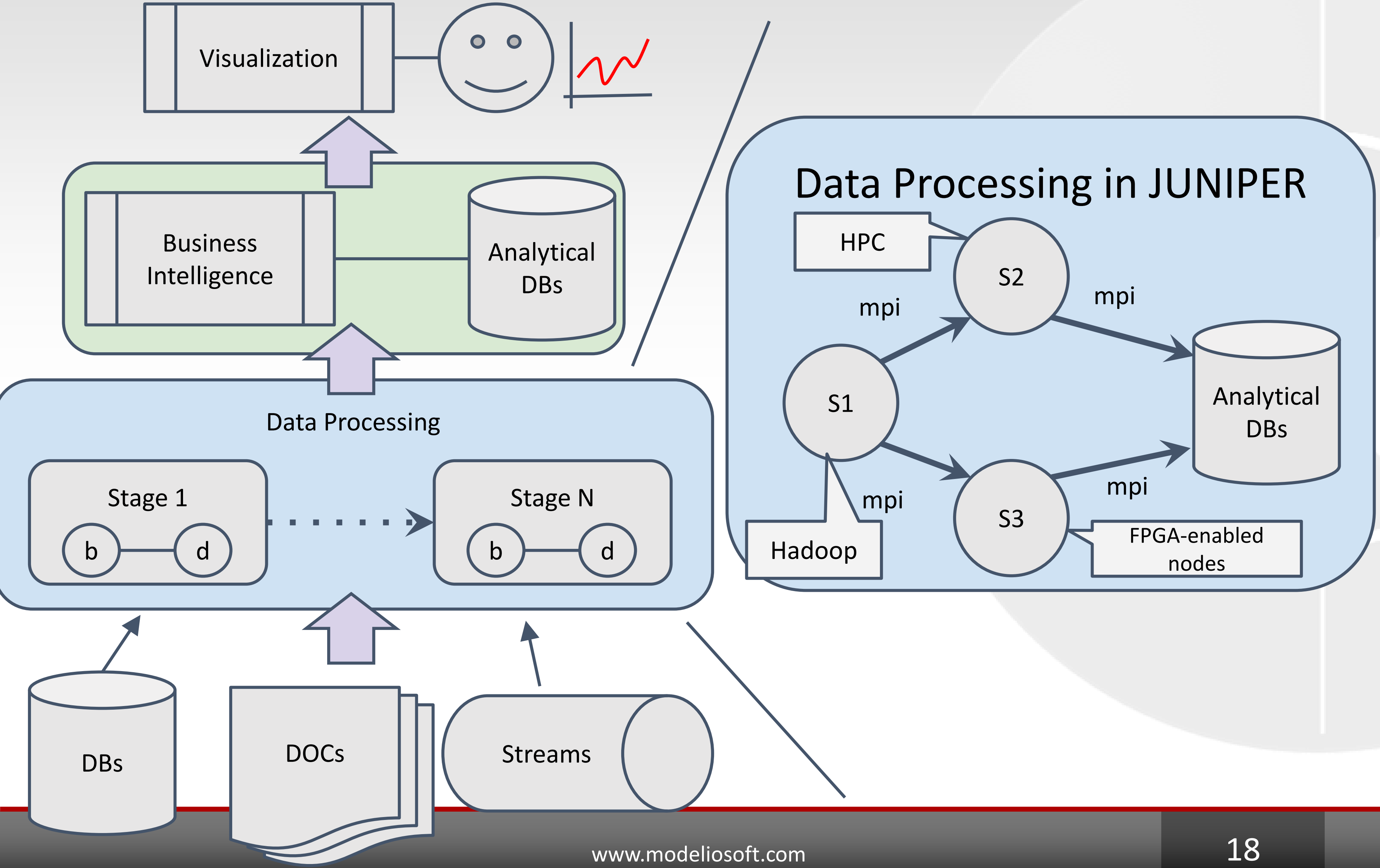
MPI for high performance computing is an option when you have a lots of money and a PhD



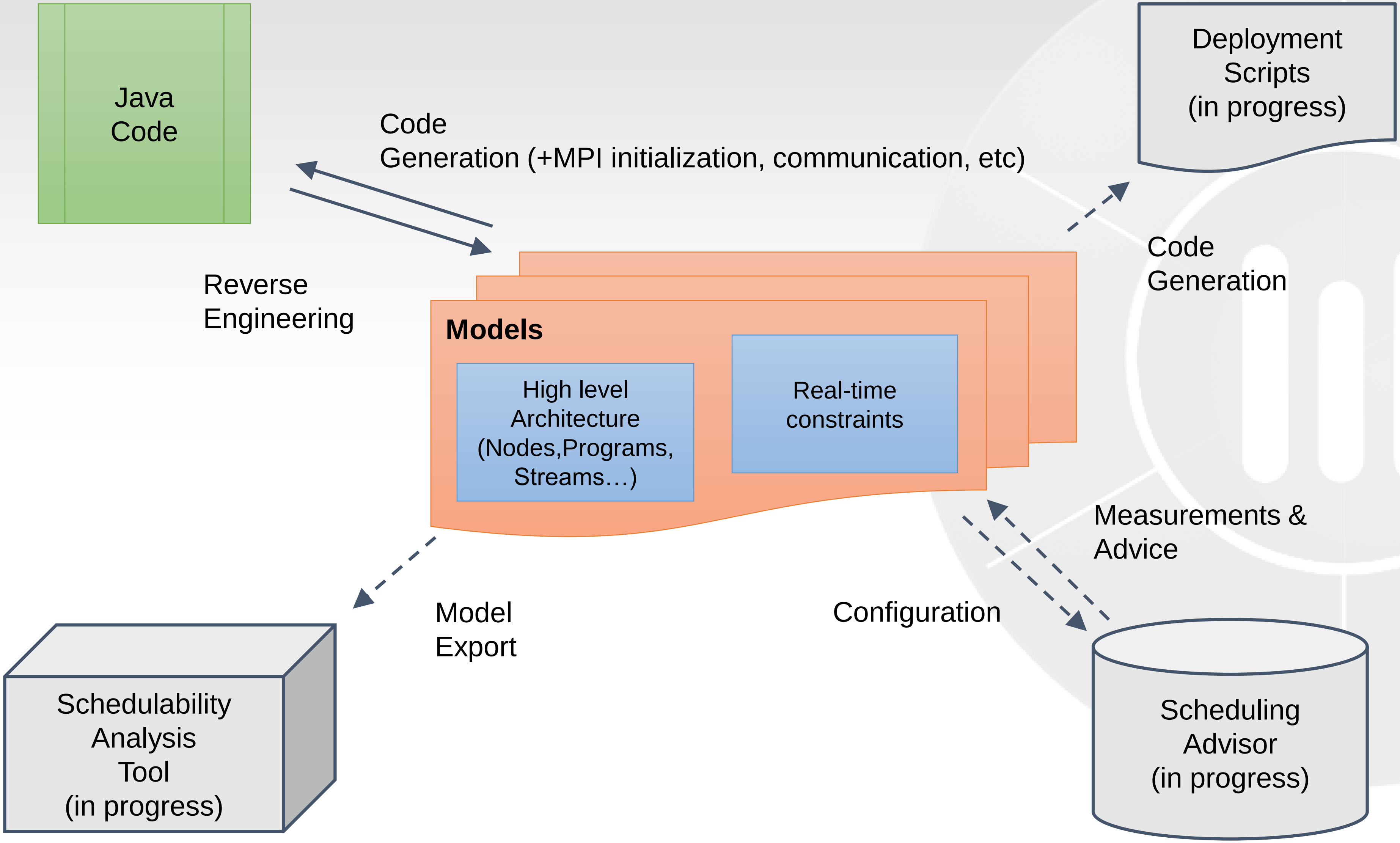
JUNIPER

JAVA PLATFORM FOR HIGH-PERFORMANCE AND REAL-TIME LARGE SCALE DATA

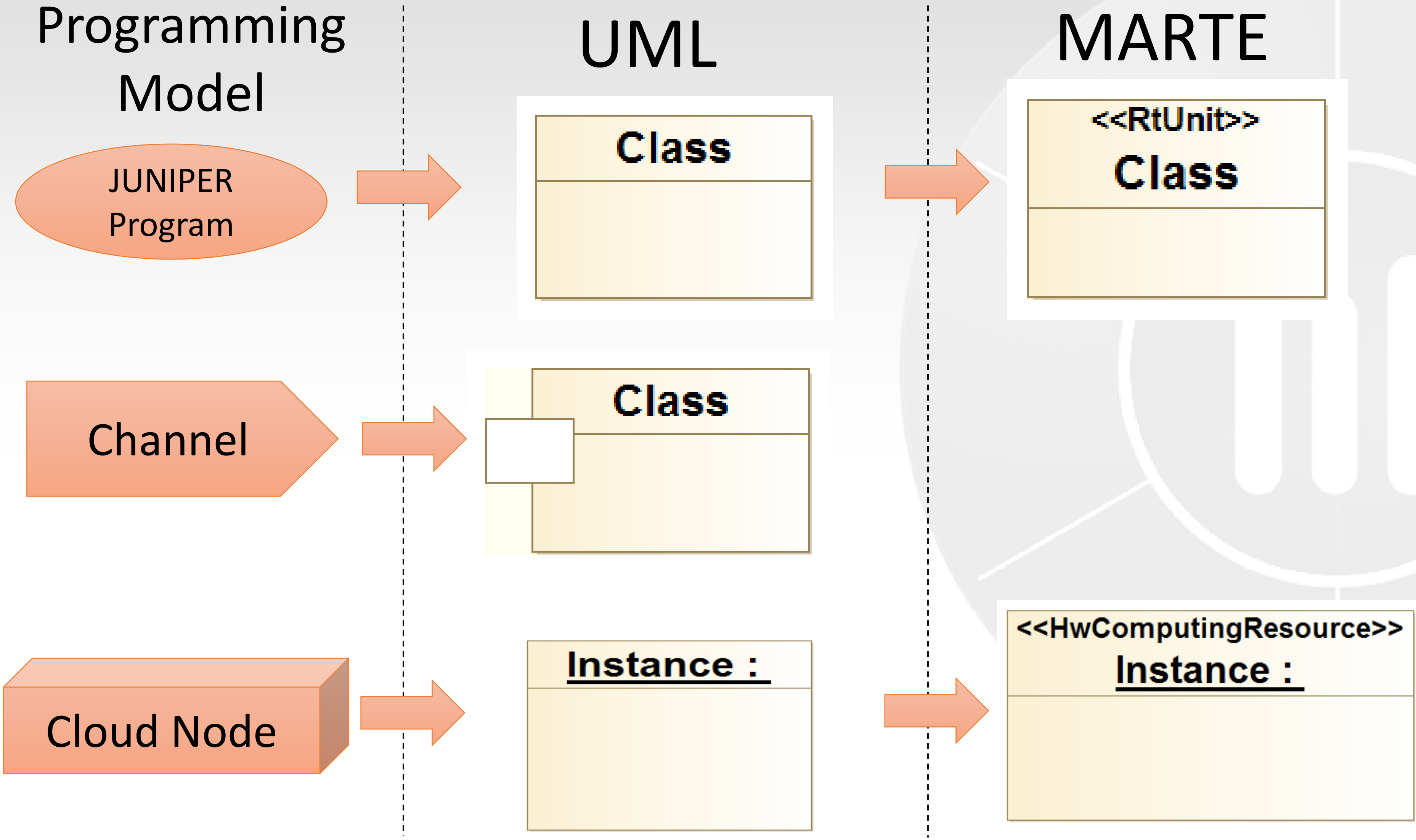
JUNIPER integrates Big Data technologies over MPI



Modelling in Juniper


Java Code
Code Generation (+MPI initialization, communication, etc)
Reverse Engineering
Deployment Scripts (in progress)
Code Generation
Models
High level Architecture (Nodes,Programs, Streams…)
Real-time constraints
Measurements & Advice
Configuration
Model Export
Schedulability Analysis Tool (in progress)
Scheduling Advisor (in progress)

Mapping Programming Model, UML and MARTE



Programming Model

UML

MARTE

JUNIPER Program

Class

<<RtUnit>>
Class

Channel

Class

Cloud Node

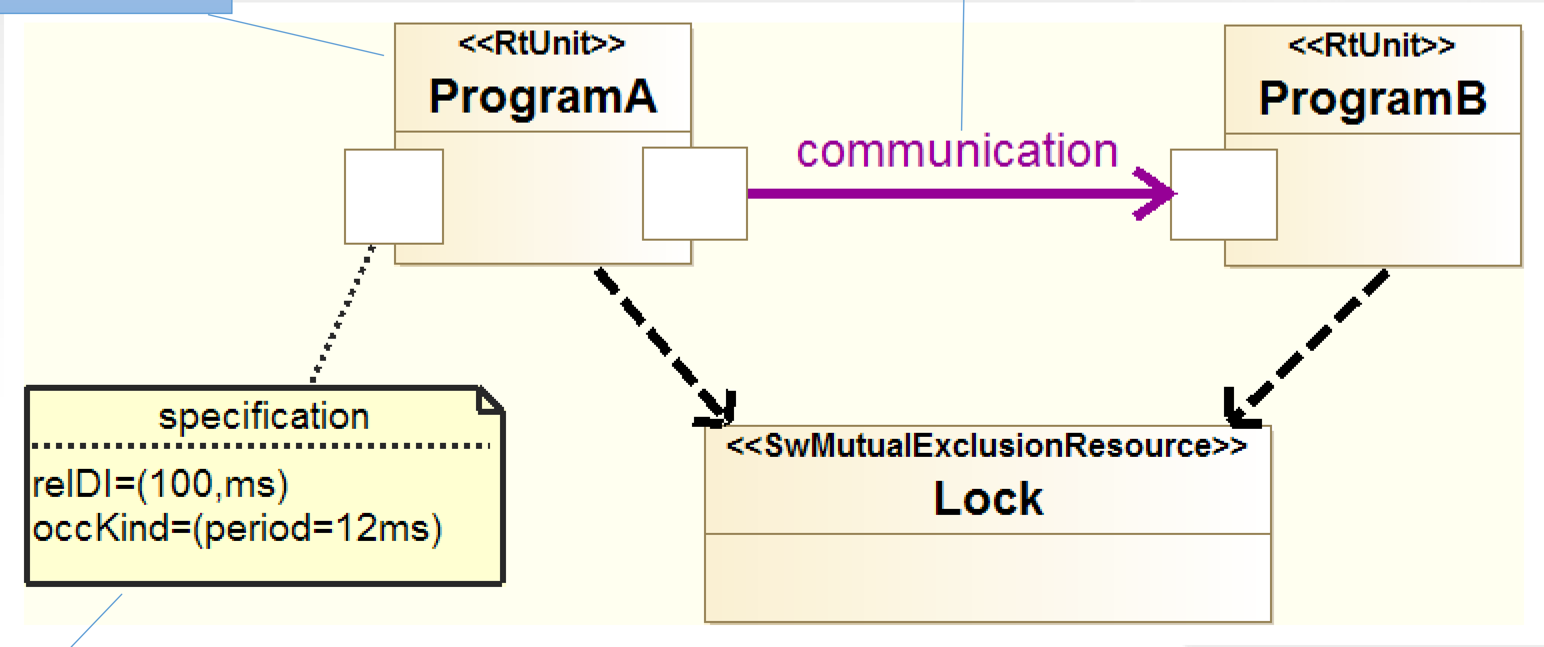
Instance :

<<HwComputingResource>>
Instance :

Modelling the application and real-time constraints

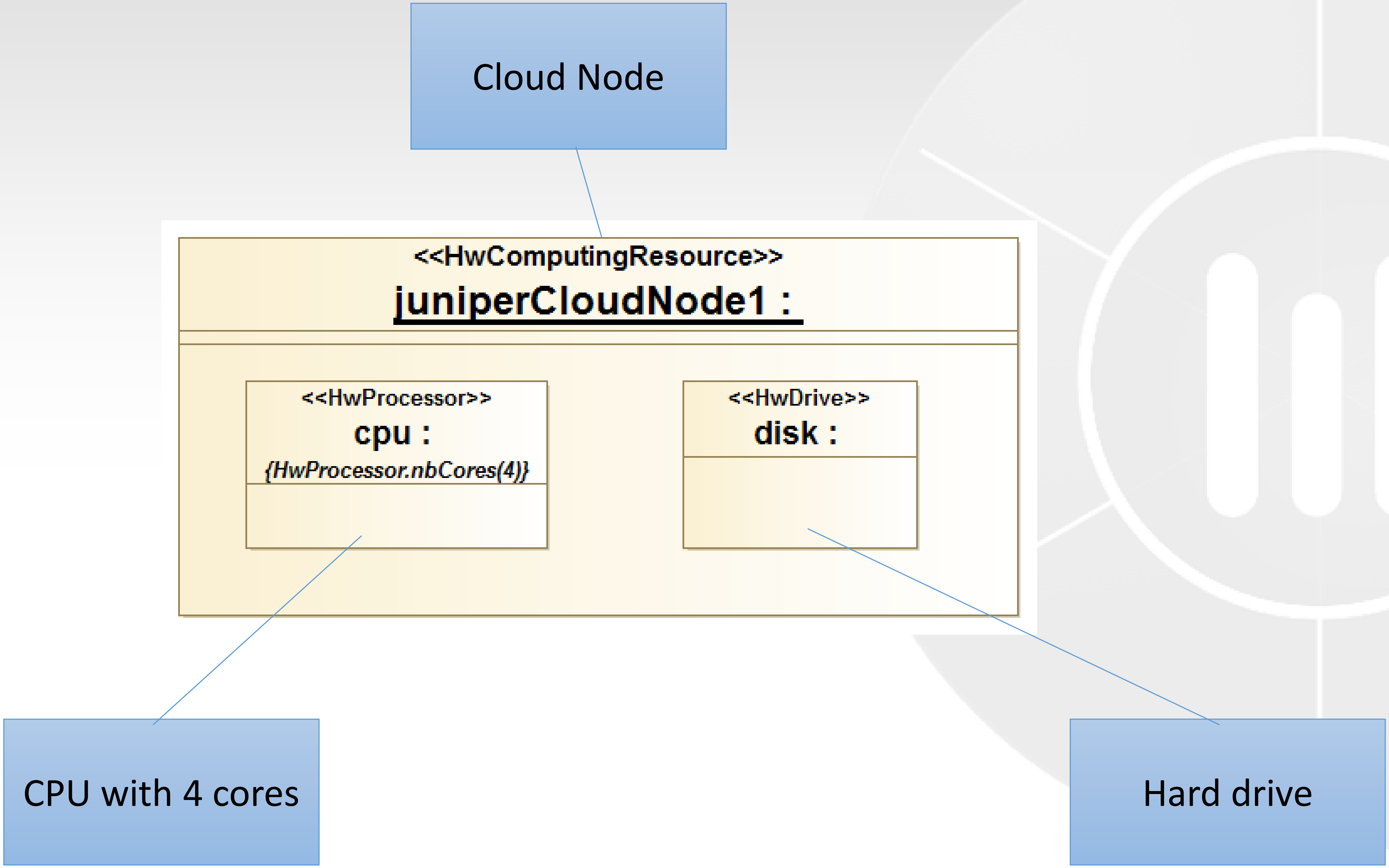
JUNIPER Programs

Big Data flow



Real-time constrains
- response time
- bandwidth

Modelling the hardware infrastructure at a high level

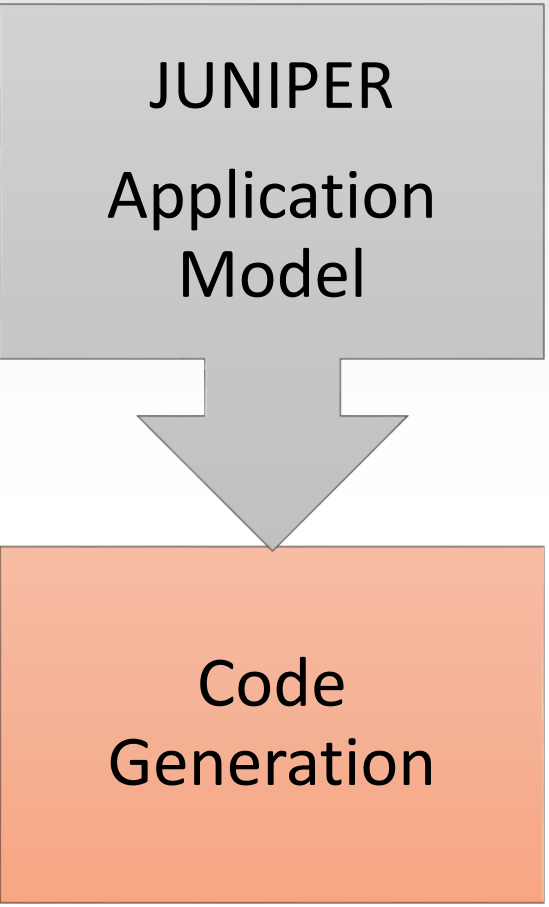


Cloud Node

CPU with 4 cores

Hard drive

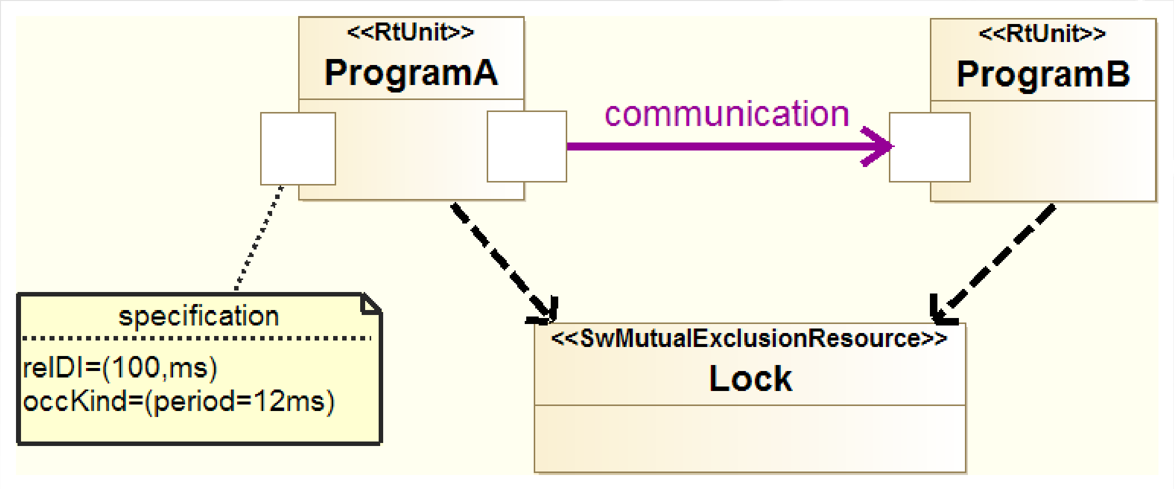
MPI code generation

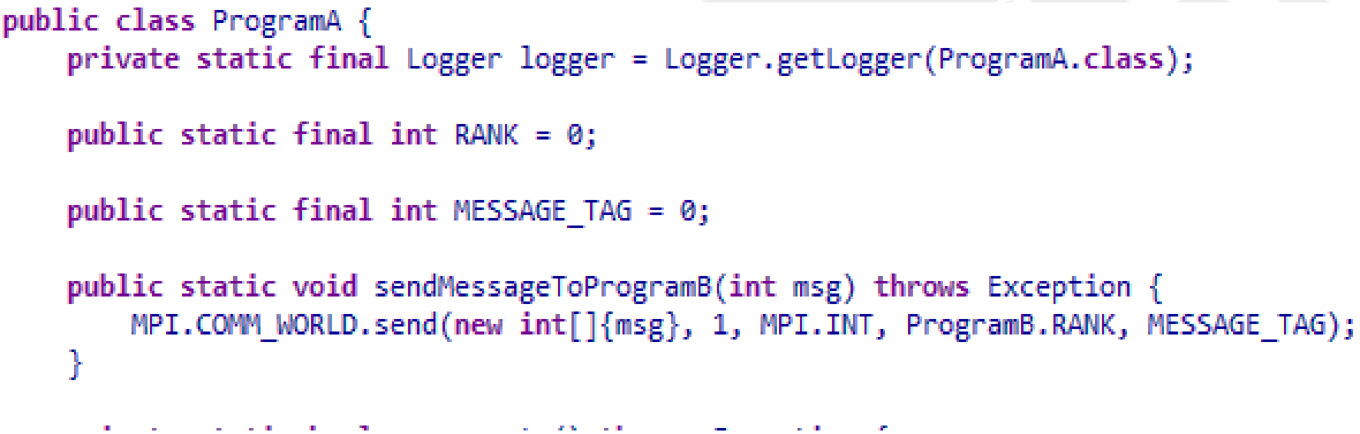


JUNIPER

Application Model

Code Generation

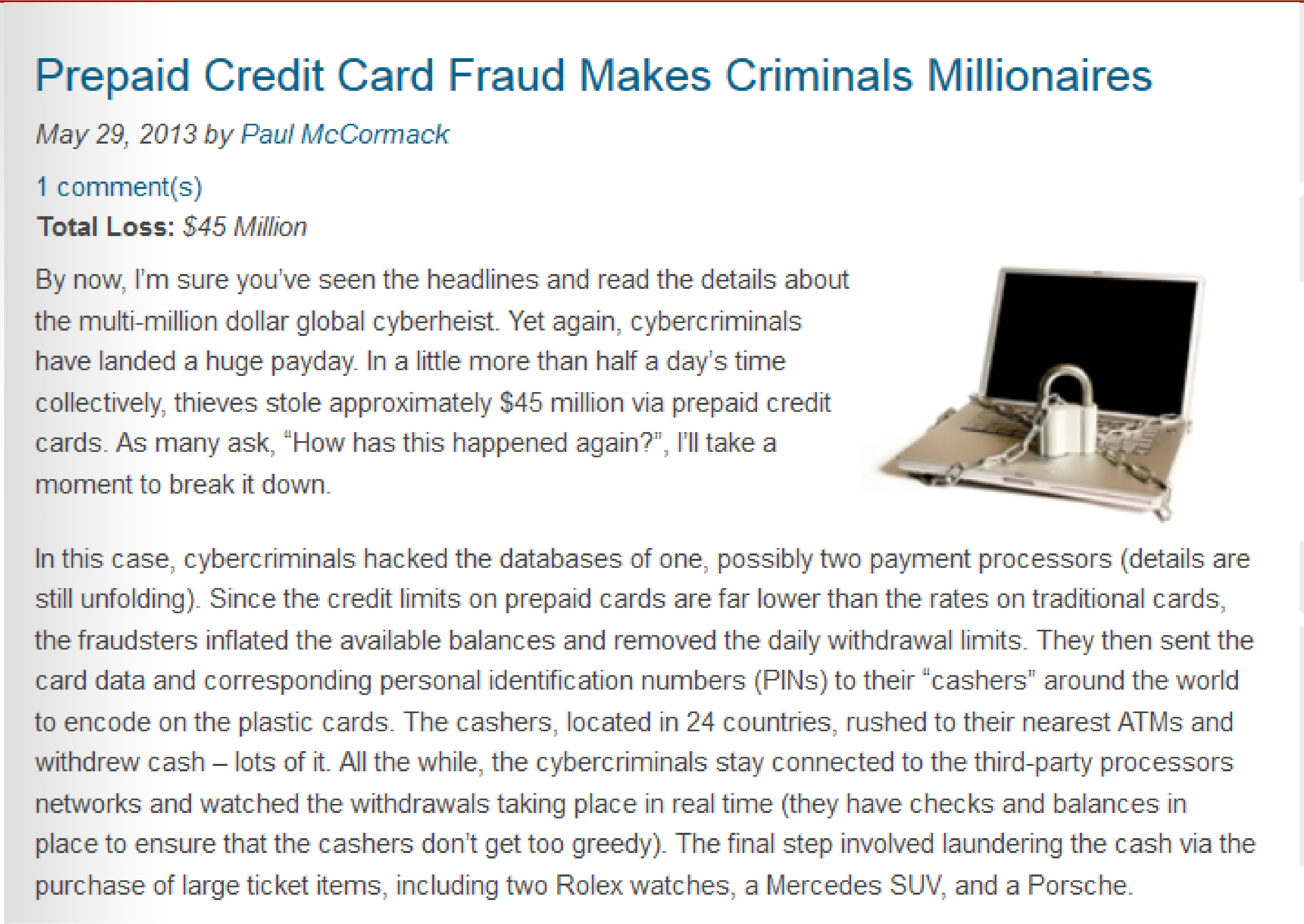




```
public class ProgramA {
    private static final Logger logger = Logger.getLogger(ProgramA.class);

    public static final int RANK = 0;

    public static final int MESSAGE_TAG = 0;

    public static void sendMessageToProgramB(int msg) throws Exception {
        MPI.COMM_WORLD.send(new int[]{msg}, 1, MPI.INT, ProgramB.RANK, MESSAGE_TAG);
    }
```

PETAFUEL CASE STUDY

Risk: $45 million in half day



Prepaid Credit Card Fraud Makes Criminals Millionaires

May 29, 2013 by Paul McCormack

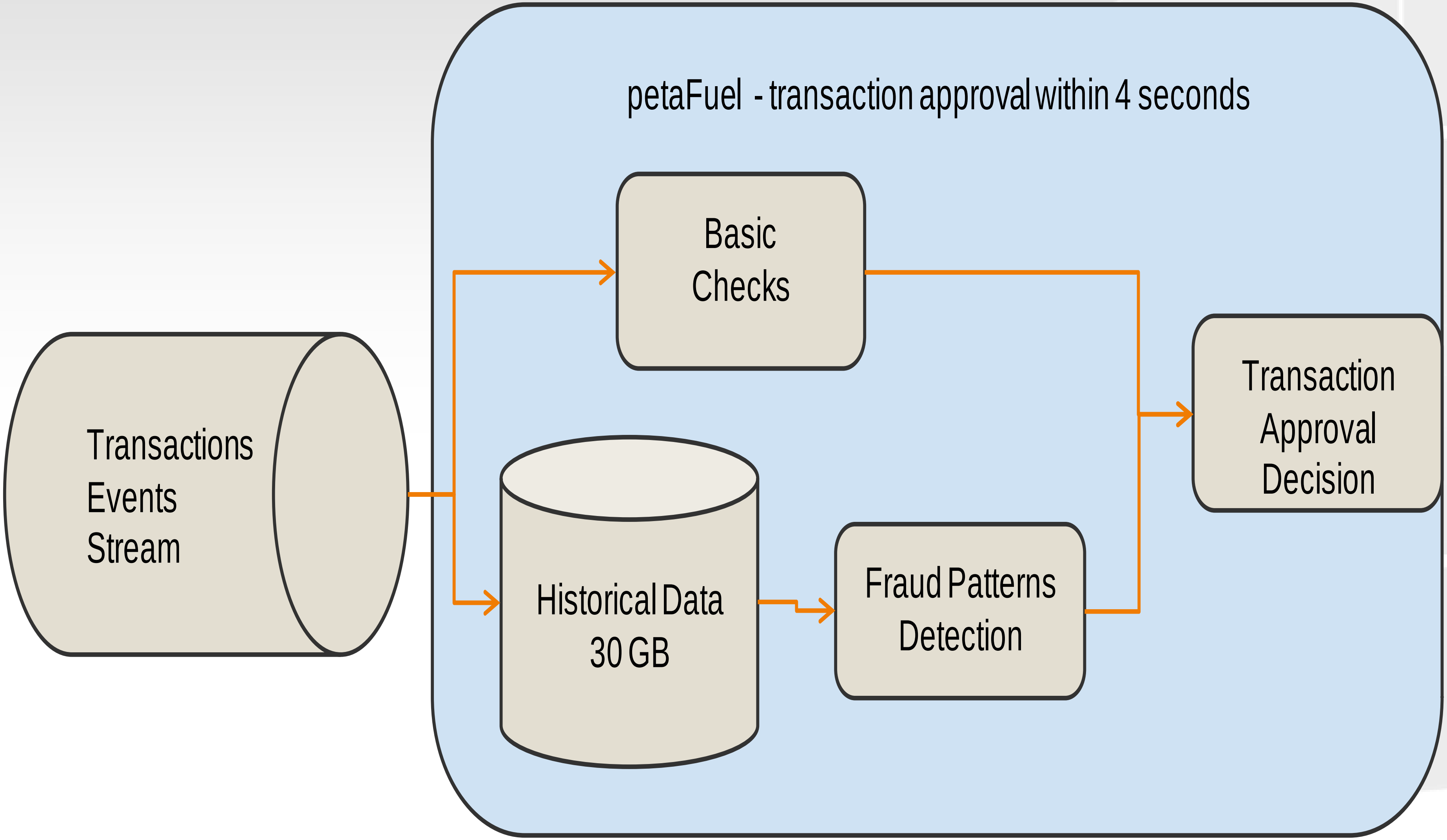
1 comment(s)

Total Loss: *$45 Million*

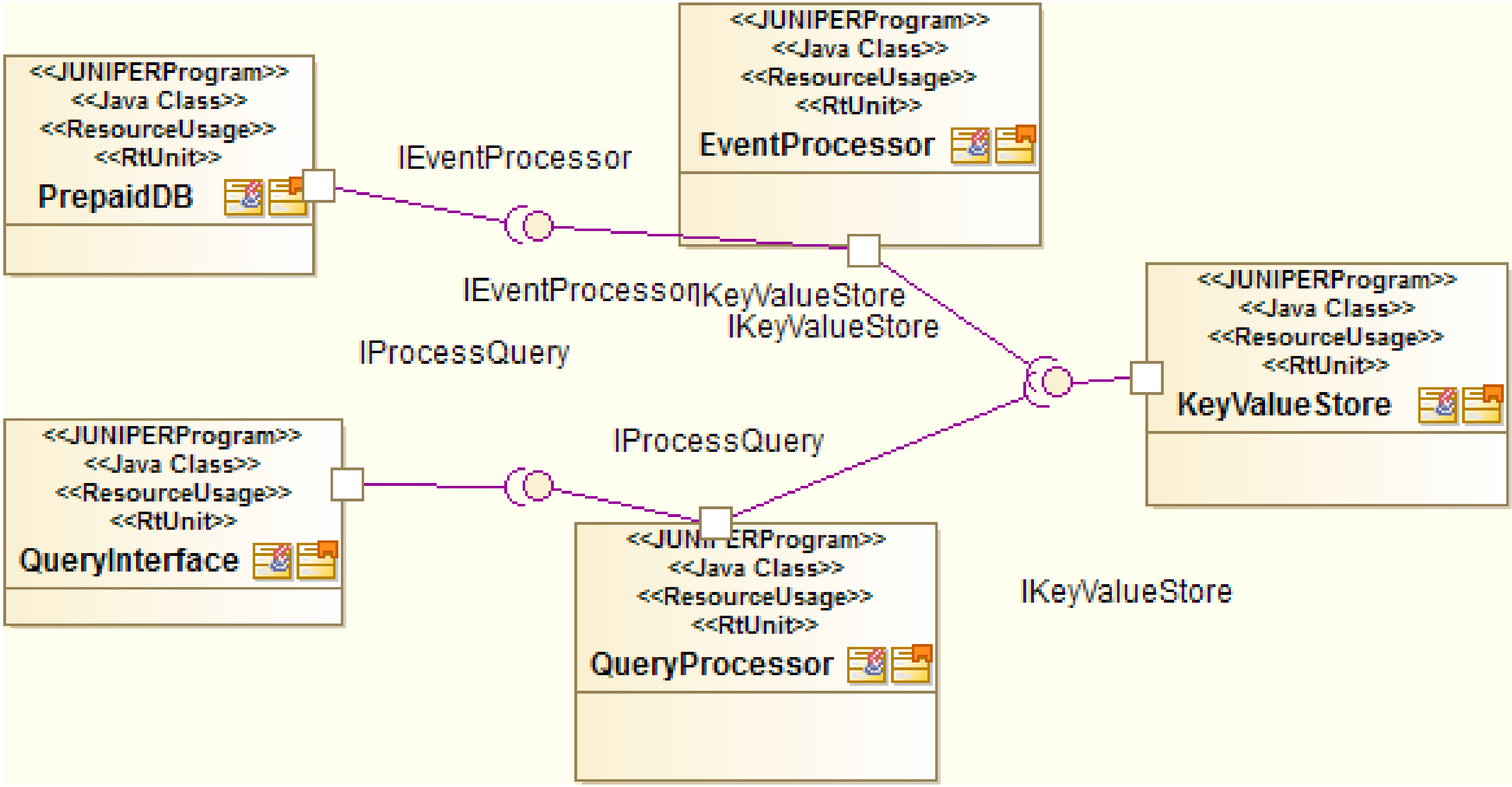
By now, I'm sure you've seen the headlines and read the details about the multi-million dollar global cyberheist. Yet again, cybercriminals have landed a huge payday. In a little more than half a day's time collectively, thieves stole approximately $45 million via prepaid credit cards. As many ask, "How has this happened again?", I'll take a moment to break it down.

In this case, cybercriminals hacked the databases of one, possibly two payment processors (details are still unfolding). Since the credit limits on prepaid cards are far lower than the rates on traditional cards, the fraudsters inflated the available balances and removed the daily withdrawal limits. They then sent the card data and corresponding personal identification numbers (PINs) to their "cashers" around the world to encode on the plastic cards. The cashers, located in 24 countries, rushed to their nearest ATMs and withdrew cash – lots of it. All the while, the cybercriminals stay connected to the third-party processors networks and watched the withdrawals taking place in real time (they have checks and balances in place to ensure that the cashers don't get too greedy). The final step involved laundering the cash via the purchase of large ticket items, including two Rolex watches, a Mercedes SUV, and a Porsche.

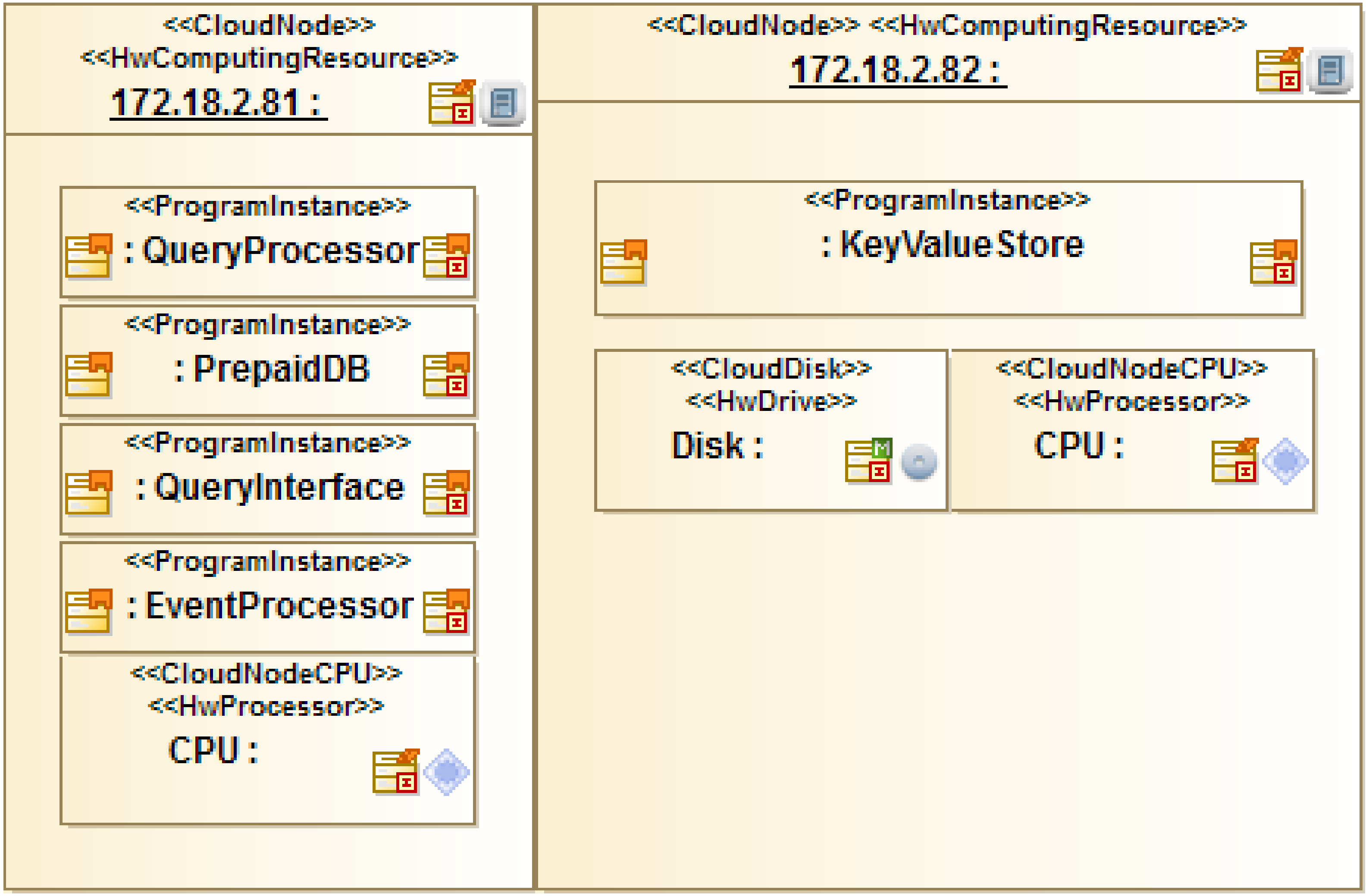
Master Card debit card approval within 4 sec


petaFuel - transaction approval within 4 seconds
Basic Checks
Transactions Events Stream
Historical Data 30 GB
Fraud Patterns Detection
Transaction Approval Decision

Juniper application model



Deployment model



MPI code generation

```
public class EventProcessor {
    public static final int RANK = 1;
    public static IEventProcessor iEventProcessorImpl = new IEventProcessor() {
        @Override
        public void process(Event event) {
            String key = getKeyFromTimestamp(event.getTimestamp());
            String value = keyValueStoreIKeyValueStore.find(key);
            if (value == null) {
                keyValueStoreIKeyValueStore.put(key, "1");
            } else {
                int count = Integer.parseInt(value);
                keyValueStoreIKeyValueStore.put(key, ""+(count+1);
            }
        }
        …
   };
    …
    public static void main(final String[] args) {
        MPI.Init(args);
        …
        MPI.Finalize();
    }
   …
}
```

CONCLUSIONS

Juniper trade-offs

	JUNIPER	
Criteria	**Hadoop**	**MPI**
Communication	HDFS - file system (httpd)	HPC cluster interconnect (Infiniband)
Data flow	Map Reduce	Modeling + MPI comms
Parallelization	Automatic	Manually based on domain decomposition
Response time guaranties	None	Real-time for single node
Stages in multi-format	No	Any (incl. Hadoop + FPGA)
Hardware	Commodity cluster	HP cluster
Price	€	€€€
Skills	+	++++
Customers	General audience	Critical systems

Prospects – more work

Work in progress

UML based language

- MPI Communication
- Timing properties
- Deployment

petaFuel case study

Future work

Modelling payload

Integrating schedulability

Running final evaluations

Final release

Questions?

*for your questions

2014
CEE-SEC(R)
Software Engineering
Conference in Russia

Andrey Sadovykh

Marcos Almeida

SOFTEAM | ModelioSoft

{name.surname}@softeam.fr

SOFTEAM R&D Web Site:

http://rd.softeam.com

Modelio Web Site :

http://www.modelio.org

JUNIPER Web Site :

http://www.juniper-project.org