



JYVÄSKYLÄN YLIOPISTO
UNIVERSITY OF JYVÄSKYLÄ

САНКТ-ПЕТЕРБУРГСКИЙ
ГОСУДАРСТВЕННЫЙ
ПОЛИТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ



Science 2.0 & Clouds for Better Research Collaboration: A Case of MIR

Evgeny Pyshkin *speaker*

*Ph.D., Senior Associate Professor, Dept. of Computer Systems and Software Engineering
St. Petersburg State Polytechnic University*

Andrey Kuznetsov

*Senior Software Engineer, Motorola Solutions Inc.,
Assistant Professor, St. Petersburg State Polytechnic University*

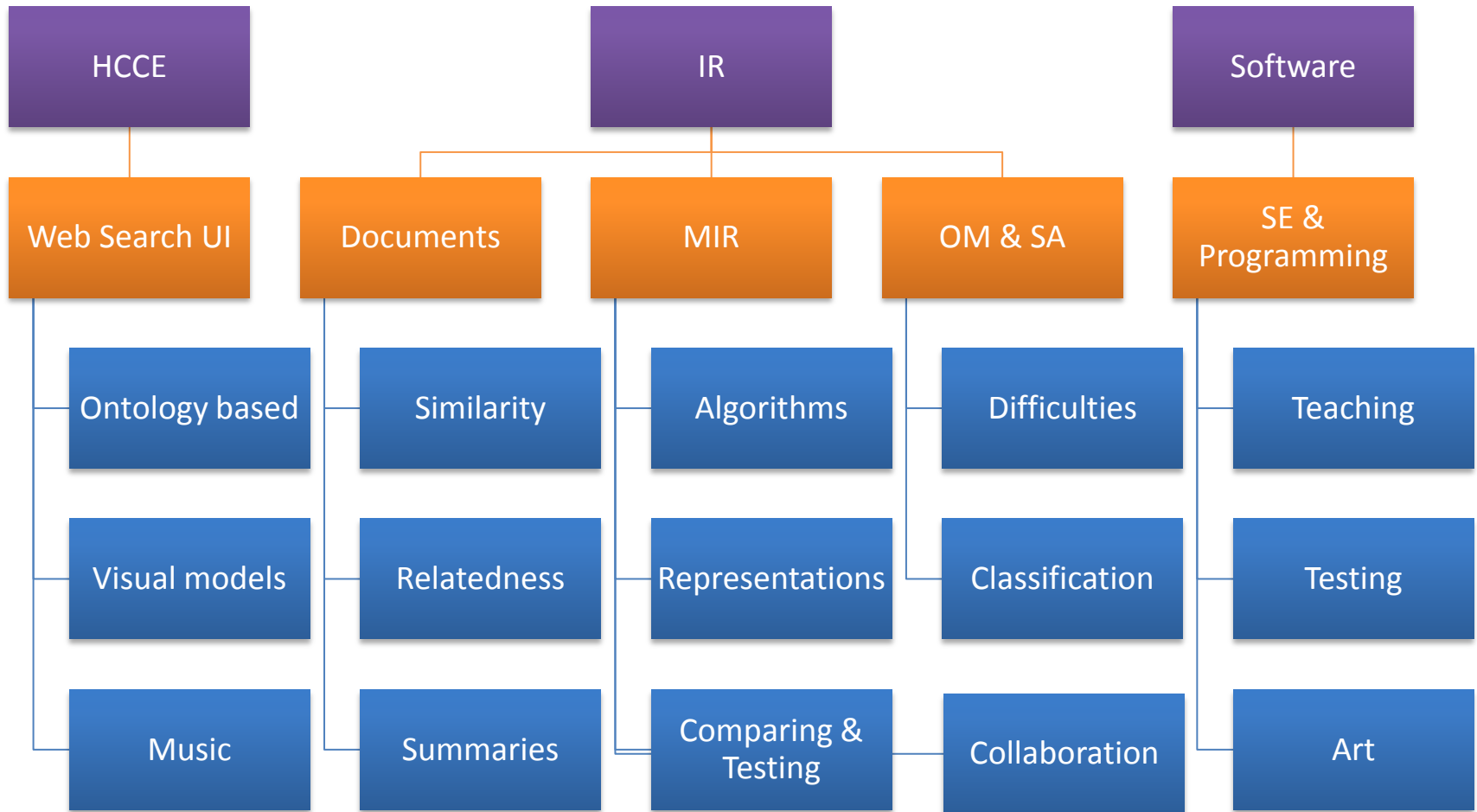
Outline

- Clouds for Better Collaboration:
A Case of MIR
 - Personal Experience
 - People and Communities
 - Ideas and Evaluations
 - Communication and Collaboration
 - Some Technical Details

Science 2.0 & Clouds
for Better Research Collaboration:
A Case of MIR

INTRODUCTION TO OUR RESEARCH PROJECTS IN THE DOMAIN OF IR (& RELATED)

Research Topics



Searching for music: from melody in mind to the resources on the web

- Improving an EMD algorithm to compare user's melody to polyphonic fragment
- Minimizing false positive results

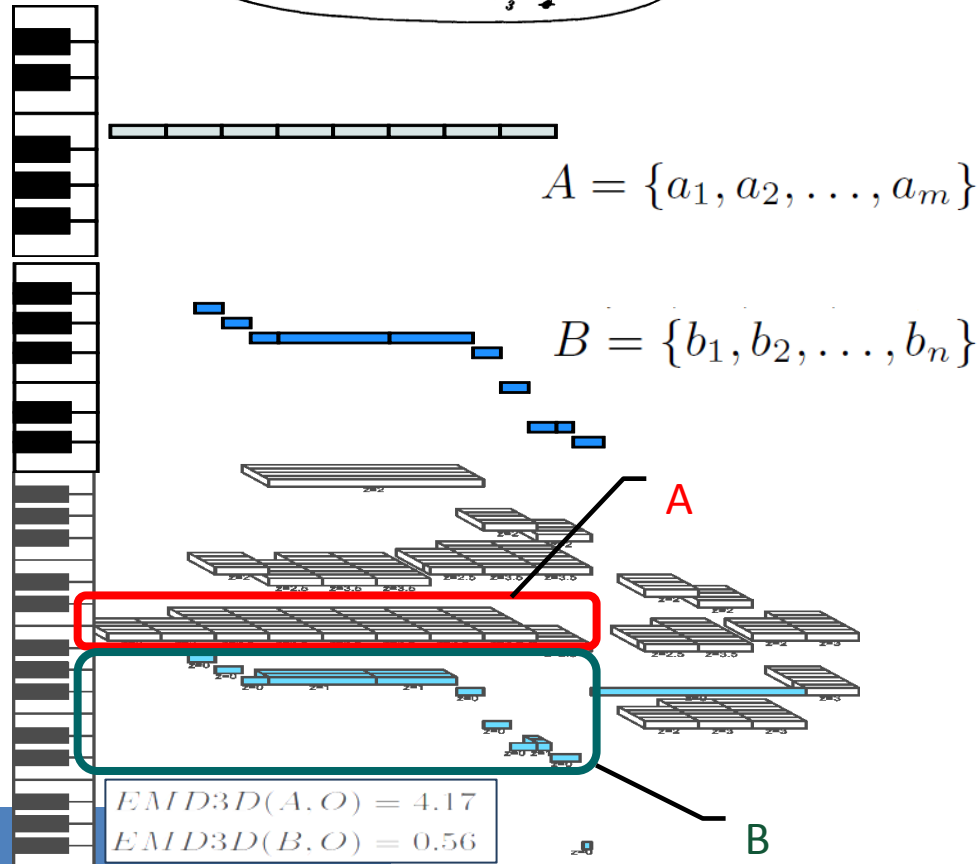
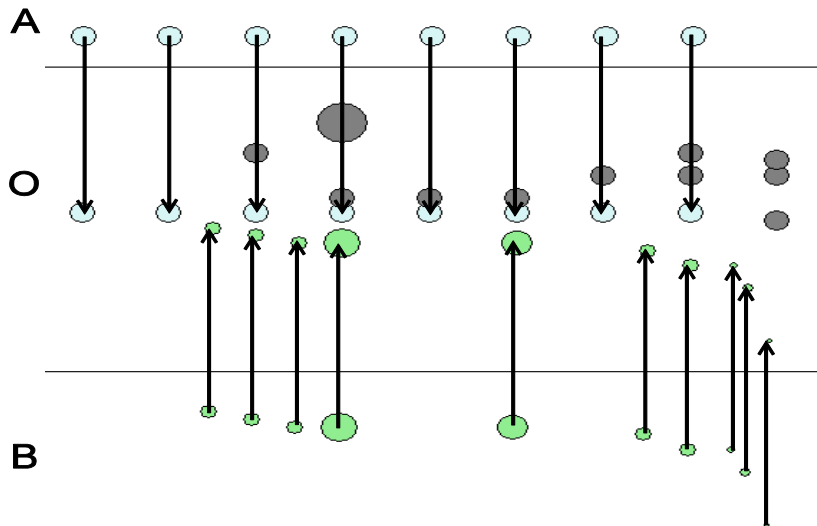


Illustration: Beethoven's Moonlight sonata



Si deve suonare tutto questo pezzo delicatissimamente e senza sordini

sempre pianissimo e senza sordini

The image shows the first four measures of the Moonlight Sonata in F# major, Op. 27, No. 1. The score is written for piano in treble and bass clefs. The treble clef part features a continuous eighth-note melody with fingerings 1, 3, 5, 1, 3, 5, 1, 2, 4, 3, 5, 1, 2, 5, 3, 5, 4, 5, 3, 5. The bass clef part consists of sustained chords with fingerings 5, 4, 5. The tempo and dynamics are marked as 'sempre pianissimo e senza sordini'.

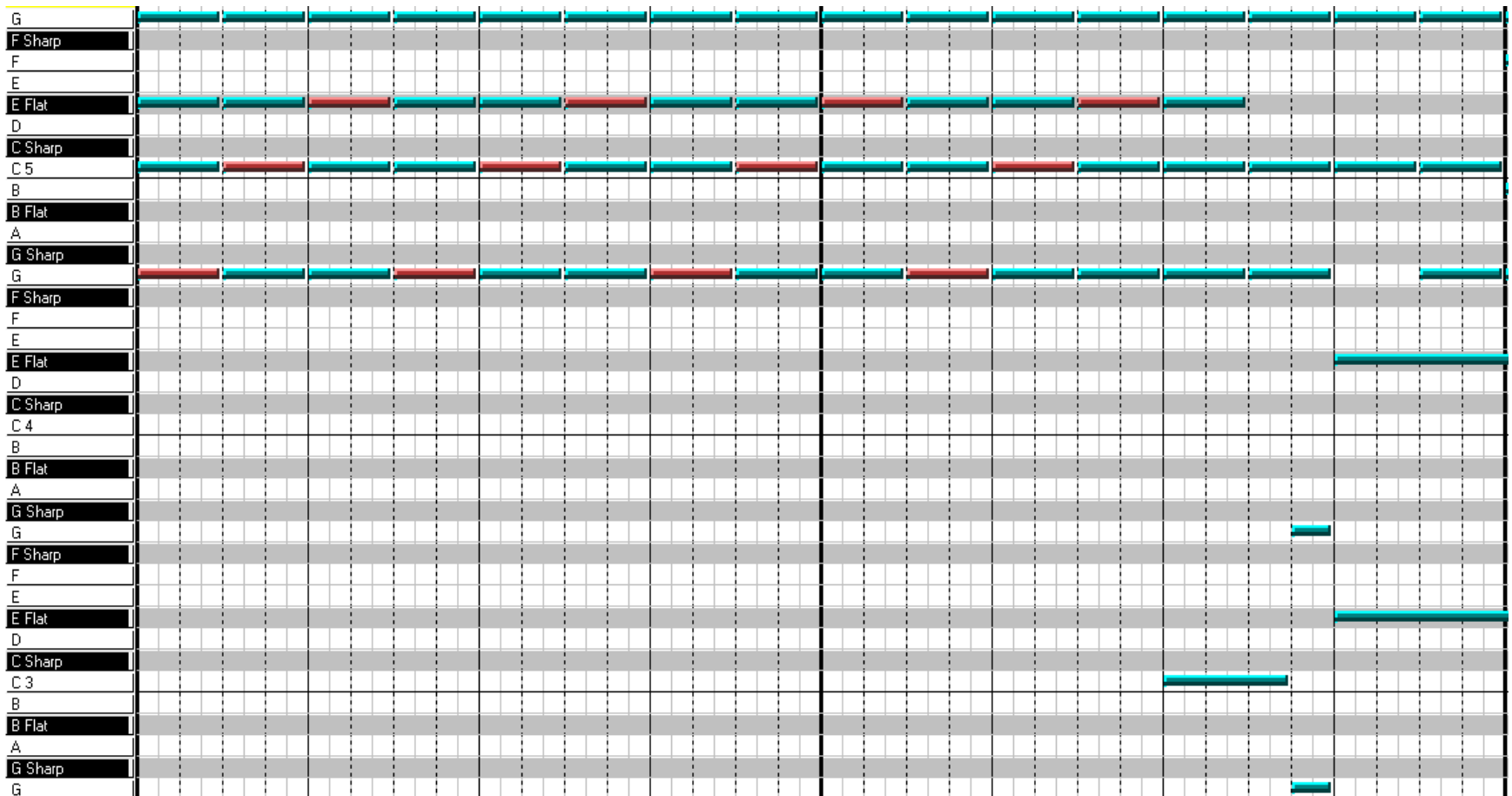
Chopin's Polonaise c-moll

Op. 40 No.2

The piano roll visualization shows the pitch contour of Chopin's Polonaise c-moll, Op. 40 No. 2. The vertical axis represents pitch, with 24 staves labeled from G to G. The horizontal axis represents time, divided into measures by vertical dashed lines. A red box highlights a section of the piano roll, specifically the staves from E Flat to F Sharp in the first few measures. A portrait of Frédéric Chopin is overlaid on the lower-left portion of the piano roll grid.



Moonlight => Chopin's Polonaise Op. 40 No.2

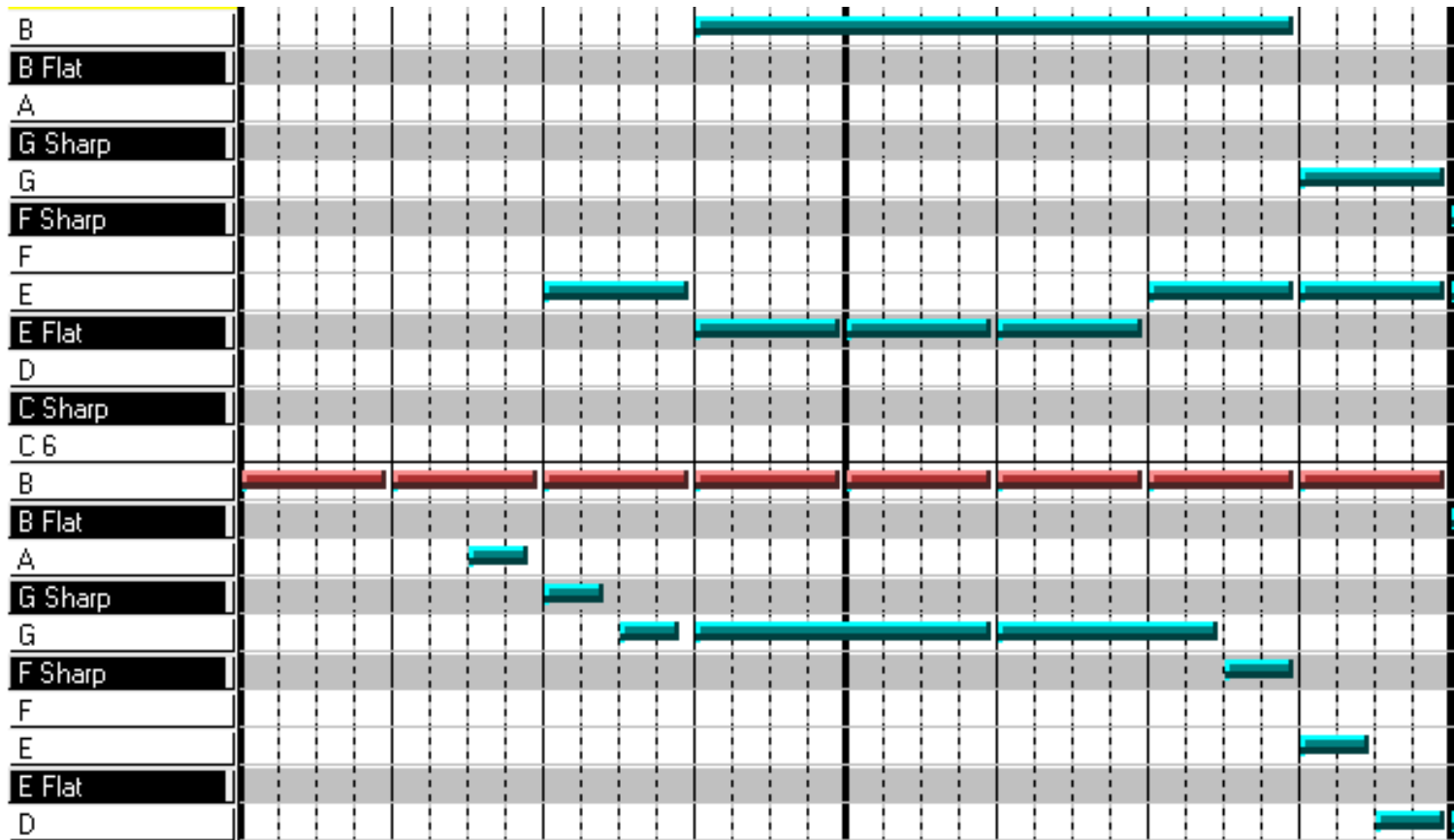


Chopin's Polonaise c-moll

Op. 40 No.2

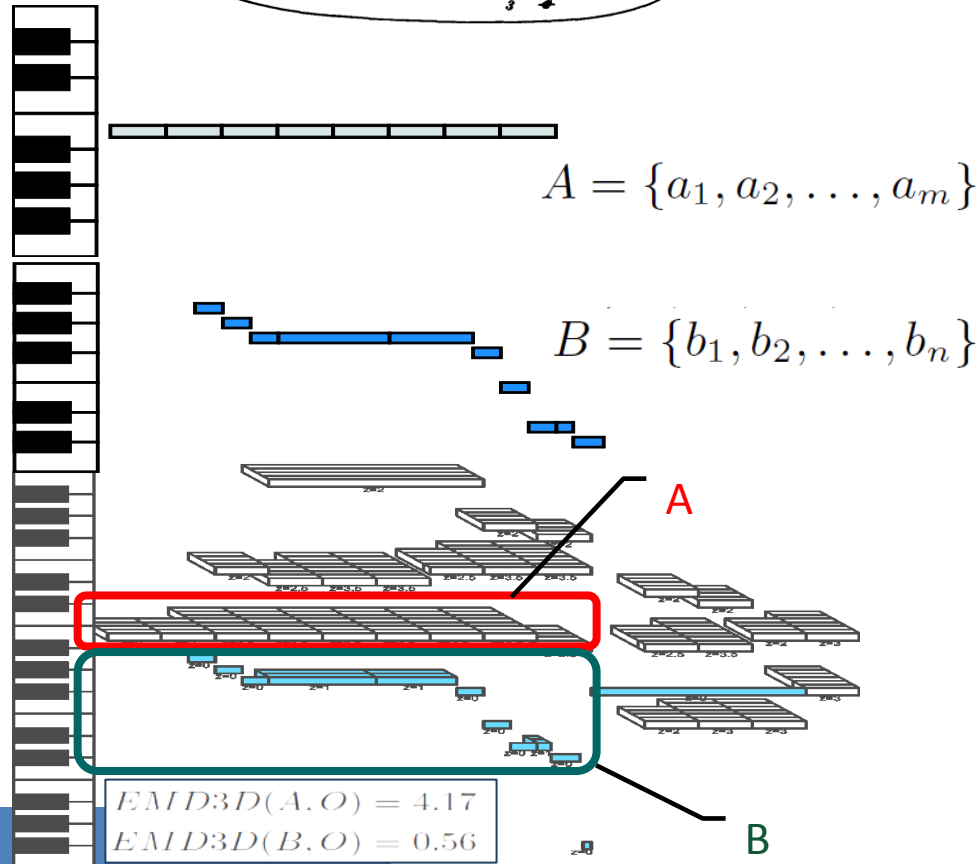
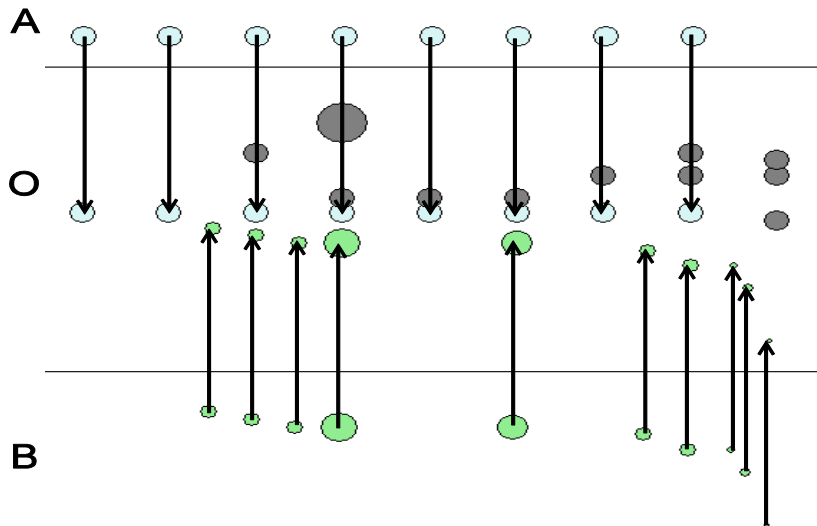


Chopin's Polonaise Op. 40 No.2 => Scriabin's Prelude No. 4, op.11



Searching for music: from melody in mind to the resources on the web

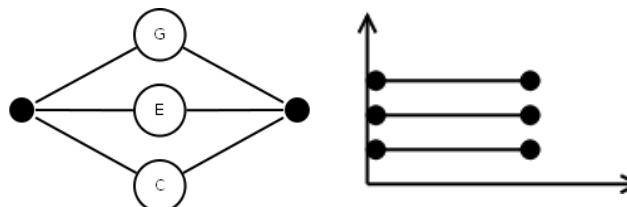
- Improving an EMD algorithm to compare user's melody to polyphonic fragment
- Minimizing false positive results



Function-based and circuit-based symbolic music representation, or Back to Beethoven

Models of Symbolic Music Representations

$\{\{C4, 1/4\},$
 $\{E4, 1/4\},$
 $\{G4, 1/4\}\}$



Function-based music representation

Si deve suonare tutto questo pezzo delicatissimamente e senza sordini

sempre pianissimo e senza sordini

$$\begin{aligned}
 F(t) = & (O(C\#2) \cdot CON(t, 0, 1, 1, 2) + \\
 & + A(t, CR(C\#4, moll, ton) \cdot Inv(0, 0, -1)) \cdot m(0, 1) \cdot v_3) + \\
 & + (O(H1) \cdot CON(t, 1, 1, 1, 2) + \\
 & + A(t, CR(C\#4, moll, ton) \cdot Inv(0, 0, -1)) \cdot m(1, 1) \cdot v_3) + \\
 & + (O(A1) \cdot CON(t, 2, \frac{1}{2}, 1, 2) + \\
 & + A(t, CR(A3, dur, ton) \cdot Inv(0, 0, 0)) \cdot m(2, \frac{1}{2}) \cdot v_3) + \\
 & + (O(F\#1) \cdot CON(t, 2\frac{1}{2}, \frac{1}{2}, 1, 2) +
 \end{aligned}$$

$$\begin{aligned}
 & + A(t, CR(D4, dur, ton) \cdot Inv(0, 0, -1)) \cdot m(2\frac{1}{2}, \frac{1}{2}) \cdot v_3) + \\
 & + (O(G\#1) \cdot CON(t, 3, \frac{1}{2}, 1, 2) + \\
 & + A(t, CR(G\#3, dur, ton) \cdot Inv(0, 0, 0)) \cdot m(3, \frac{1}{4}) \cdot v_3 + \\
 & + A(t, CR(C\#4, moll, ton) \cdot Inv(0, 0, -1)) \cdot m(3\frac{1}{4}, \frac{1}{4}) \cdot v_3) + \\
 & + (O(G\#1) \cdot CON(t, 3\frac{1}{2}, \frac{1}{2}, 1, 2) + \\
 & + A(t, CR(G\#3, moll, sus4)) \cdot Inv(0, 0, 0)) \cdot m(3\frac{1}{2}, \frac{1}{4}) \cdot v_3 + \\
 & + A(t, CR(H\#3, moll, dim) \cdot Inv(0, 0, -1)) \cdot m(3\frac{3}{4}, \frac{1}{4}) \cdot v_3)
 \end{aligned}$$

Spelling Out Opinions: Difficult Cases of Sentiment Analysis

Difficult cases:

	What groups can be identified?
A.	Indirect and Hidden Opinions
B.	Difficulties of Opinion or Sentiment Word Detection
C.	Errors Conditioned by Spelling and Grammar Factors
D.	Feature Interinfluence in Feature-based Opinion Mining
E.	Opinions Temporal Dependency
F.	Case of Multilingualism
G.	Subjective Opinions and Their Semantic Orientation

- Opinions and sentiments which are multi-faceted
- When sentiment classifiers are wrong?

Indirect and Hidden Opinions

Example:

“You'd better read the book”

- It's important to know something not only about the object of sentiment but about the entire expression.

Negative

- hardly
- possible

Positive

- extremely
- totally

Related published works

- A. Kuznetsov, and E. Pyshkin, ***“Searching for music: from melody in mind to the resources on the Web,”*** In Proceedings of the 13th International Conference on Humans and Computers (HC2010), Dec. 8-10, Aizu-Wakamatsu, Japan, 2010, 152–158.
- A. Kuznetsov, and E. Pyshkin, ***“Function-based and circuit-based symbolic music representation, or Back to Beethoven,”*** In Proceedings of the 14th International Conference on Humans and Computers, The Joint Conference on Human-Centered Computer Environment (HCCE-2012), Aizu-Wakamatsu, Japan, March, 8 – 13, 2012, 171–177. – University of Aizu Press. – 2012.
- I. Khozyainov, E. Pyshkin, and V. Klyuev. ***“Spelling out opinions: Difficult cases of sentiment analysis,”*** In Proceedings of 2013 International Joint Conference on Awareness Science and Technology & Ubi-Media Computing (iCAST-UMEDIA), Nov 2-4, Aizu-Wakamatsu, Japan, pp. 231–237.

Scientific community demands

- Computational and data resources distribution
- Ability to reproduce results reported by other researchers
- Getting access to algorithms and test collections
- Inspired* by Music Information Retrieval (MIR) domain

** See ISMIR 2012: “Reusable software and reproducibility in music informatics research”*

Science 2.0 & Clouds
for Better Research Collaboration:
A Case of MIR

SCIENCE 2.0: GETTING INVOLVED

Science 2.0 Initiative (1)

Current Model

- Research done **privately**; then submitted to journals; then peer-reviewed by gatekeepers in major journals; published
- Scientific literature **behind paywalls** online
- Credit established by **journal** name or journal impact factor.

Emerging Model

- Research **data shared** during discovery stages; ideas shared; scientists collaborate; then findings are disseminated online
- Scientific discoveries **free** online
- Credit established by citation count, number of **views** or downloads.

Science 2.0 Initiative (2)

Current Model

- Data is **private until** publication
- Papers generally protected by **copyright**
- Publishers raise funds by **charging** for access to content
- Journal article summaries available online **after publication**

Emerging Model

- Data is **shared before** publication
- Many **different licenses** possible
- Publishers seek **alternative funding**
- **Share** methods, data, findings via blogs, social networking sites, wikis, computer networking,, video journals, etc.

--- Current methods ---

Form hypothesis

Gather data privately; test

Write journal article

Submit for review

Peer-review gatekeepers?

Publish

Reject

Information available to public

--- Emerging methods ---

Form hypothesis

Share ideas, methods, data with other scientists online via blogs, video journals, social networks, and other methods

Test, perform experiments

Share findings online in preliminary form

Publish in blogs, wikis, as well as journals

Information available to the public and to other scientists as it is being developed and tested; data available too

Science 2.0: Benefits & Drawbacks

Benefits

- more collaborative
- freer, less expensive
- faster development
- wider access & diverse applications
- lets other scientists see results instantly and comment

Drawbacks

- difficulty getting paid
- risk others will copy preliminary work to get credit, patents, money
- how will reviewers and editors get paid?
- need infrastructure
- discouraging

Science 2.0 & Clouds
for Better Research Collaboration:
A Case of MIR

**AN ATTEMPT TO CONTRIBUTE:
PROVISIONING RESEARCH
APPLICATIONS IN CLOUDS**

Major Problems in (M)IR

- Access to the implementation
 - Unpublished / Legal issues
- Reproducibility
 - Implementation doesn't work properly
 - Test data aren't accessible (copyright restrictions, big size)
- Expertise
 - Third party execution within the local environment

Known Solutions

(out of scope of our approach)

- Publishing a source code as files
- Publishing a dataset as files
- Publishing completely configured VM as files
- Publishing completely configured VM as a service
 - E.g. remote workplace
- Outsourcing to third party organization
 - E.g. MIREX

Known Solutions

(in scope of our work)

- Publishing algorithms as services
 - E.g. in a cloud
- Publishing data as a services
 - E.g. in a cloud
- Reason:
 - No need to publish application nor dataset collection => no copyright issues violation

MIR Research Software in Practice

- 82 % of researchers developed software
 - Only 39% of those took steps toward reproducibility
 - Only 35% of those published any code
- Only 11% made efforts toward the reproducibility of result
- 51% said their code never left their own computer

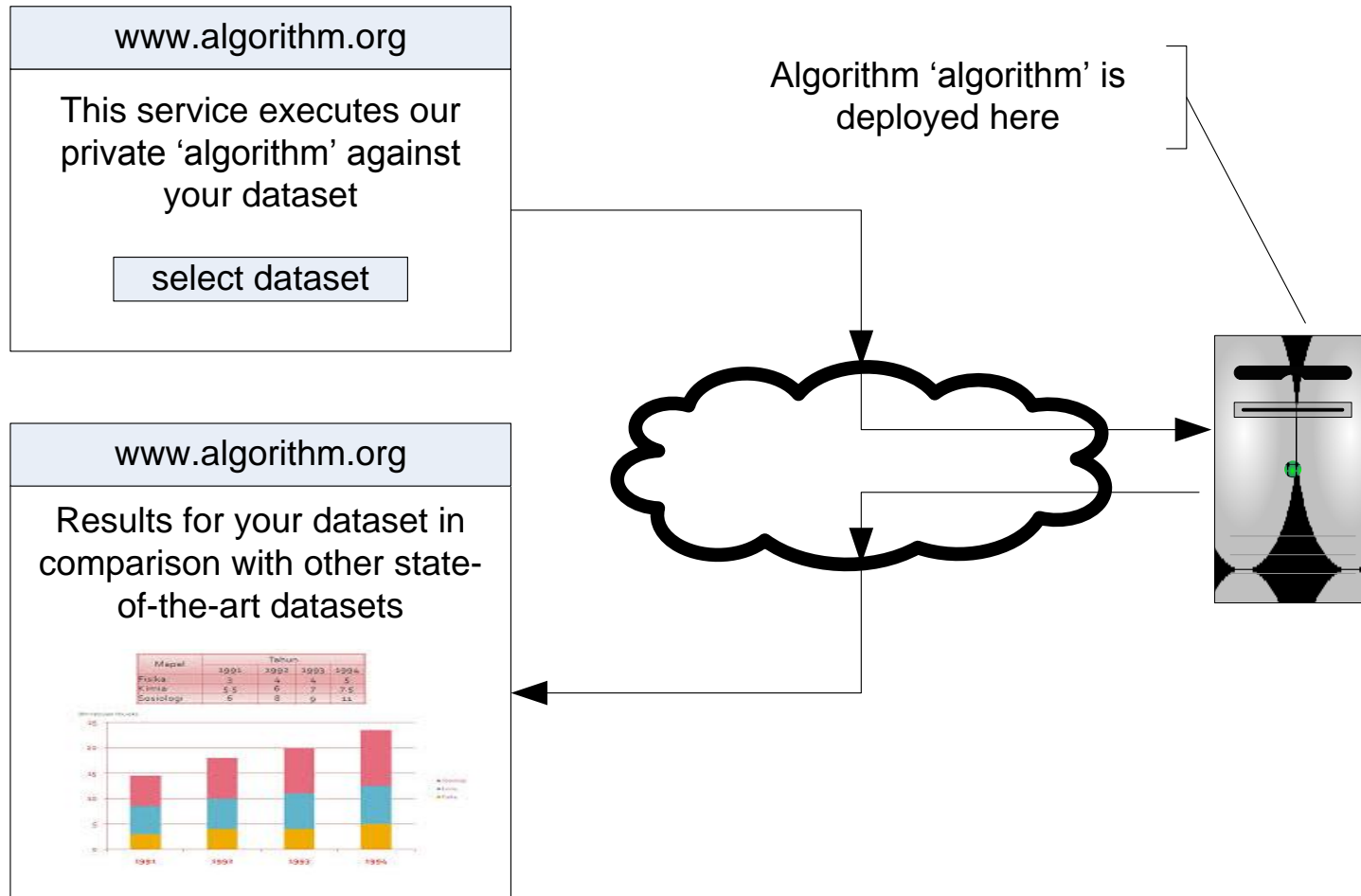
We believe in the results reported in papers without being sure that the reported results came from a certain method, and not from bugs in the software

What are difficulties? **

- There are some for IT experts
- There are many for non IT experts
 - Example: MIR community
 - Research software is developed as desktop applications not intended to be executed in networked or distributed environments
 - Researchers aren't experienced enough to resolve deployment problems and to configure cloud runtime environment properly

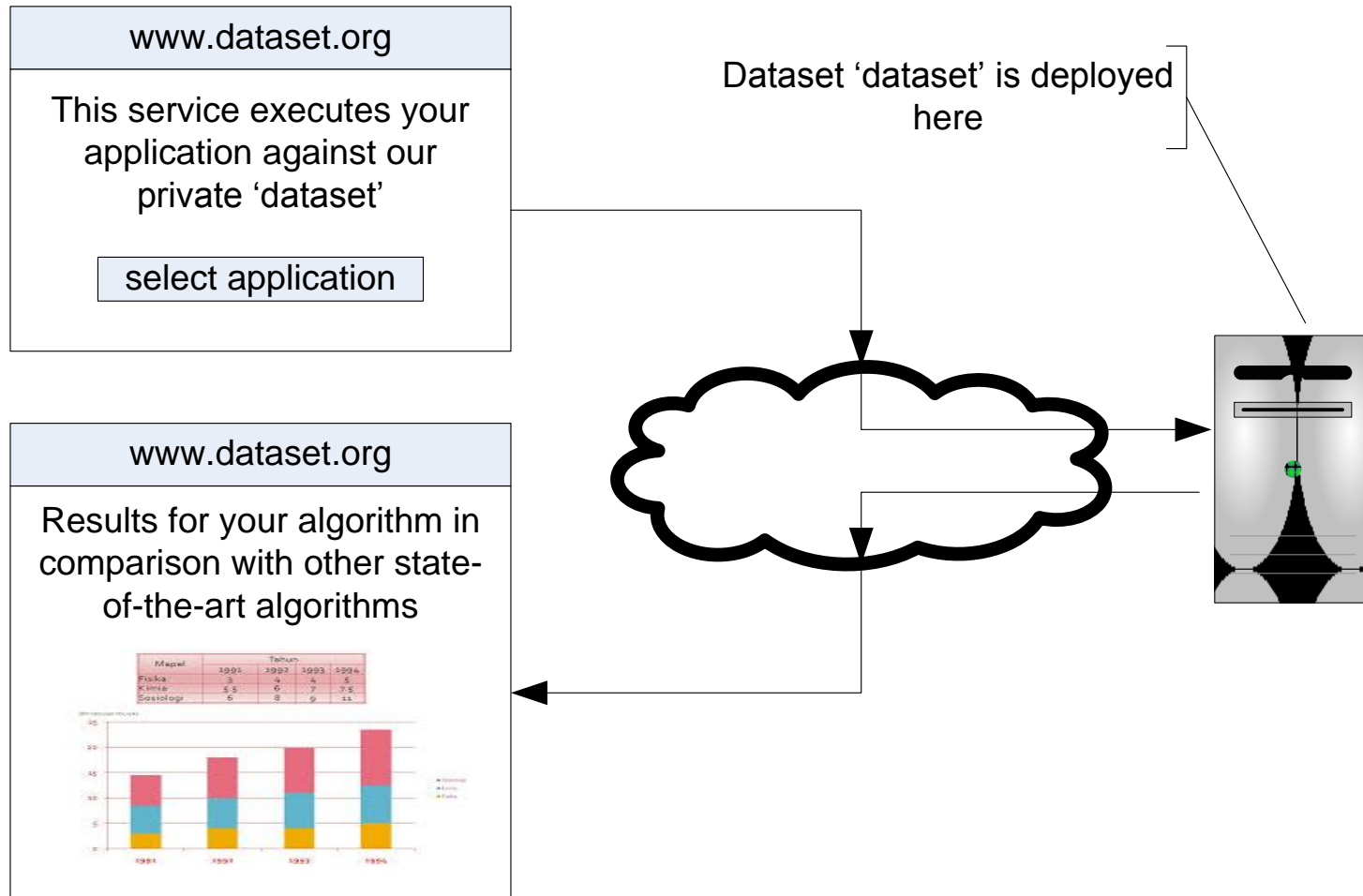
**** *to deploy applications in clouds***

Cloud in Theory (Algorithm as a Service)



Cloud in Theory

(Dataset as a Service)



Cloud in Practice (OpenShift example)

The screenshot shows a web browser window with the following tabs: "Applications | OpenShift Online", "HPCC Conference | HPCC 2014", and "conference.hpcc2014.studio". The address bar displays "https://openshift.redhat.com/app/console/applications". The browser's bookmark bar includes "Apps", "Insert title here", "JBoss", "music", "ant", "JBoss BRMS", "Dashboard [Jenkins]", "ClearCase", "amazon", "auth", and "Re: Recommend".

The main content area features the "OPENS SHIFT ONLINE" logo and a navigation menu with "Applications", "Settings", and "Support". Below the navigation, the "Applications" section is displayed, showing "1 of 3" items. The first application is "jbossas", described as "JBoss Application Server 7, PostgreSQL 9.2". A button labeled "Add Application..." is visible at the bottom of the application list.

Cloud in Practice (OpenShift example)

The screenshot shows a web browser window with the URL `https://openshift.redhat.com/app/console/application_types`. The browser tabs include "Create a New Application", "HPCC Conference | HPCC 2014", and "conference.hpcc2014.studio". The browser's address bar and tabs show various folders and files, including "JBoss BRMS", "Dashboard [Jenkins]", "ClearCase", "amazon", "auth", "Re: Recommended Pr...", and "Overlord - JBoss".

The main content area is divided into two columns: "Instant App" and "xPaaS".

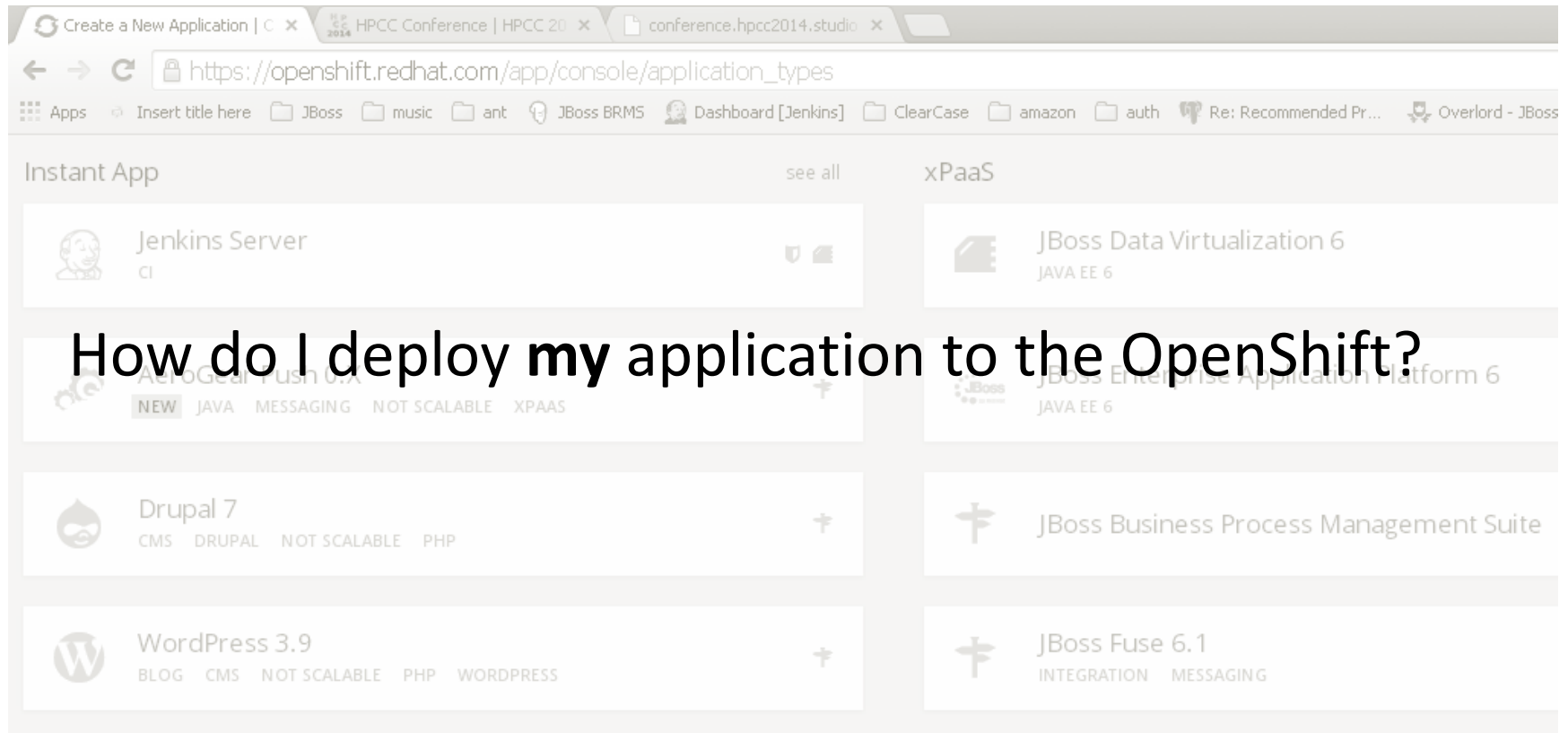
Instant App (with a "see all" link):

- Jenkins Server** (CI)
- AeroGear Push 0.X** (NEW, JAVA, MESSAGING, NOT SCALABLE, XPAAS)
- Drupal 7** (CMS, DRUPAL, NOT SCALABLE, PHP)
- WordPress 3.9** (BLOG, CMS, NOT SCALABLE, PHP, WORDPRESS)

xPaaS (with a "see all" link):

- JBoss Data Virtualization 6** (JAVA EE 6)
- JBoss Enterprise Application Platform 6** (JAVA EE 6)
- JBoss Business Process Management Suite**
- JBoss Fuse 6.1** (INTEGRATION, MESSAGING)

Cloud in Practice (OpenShift example)

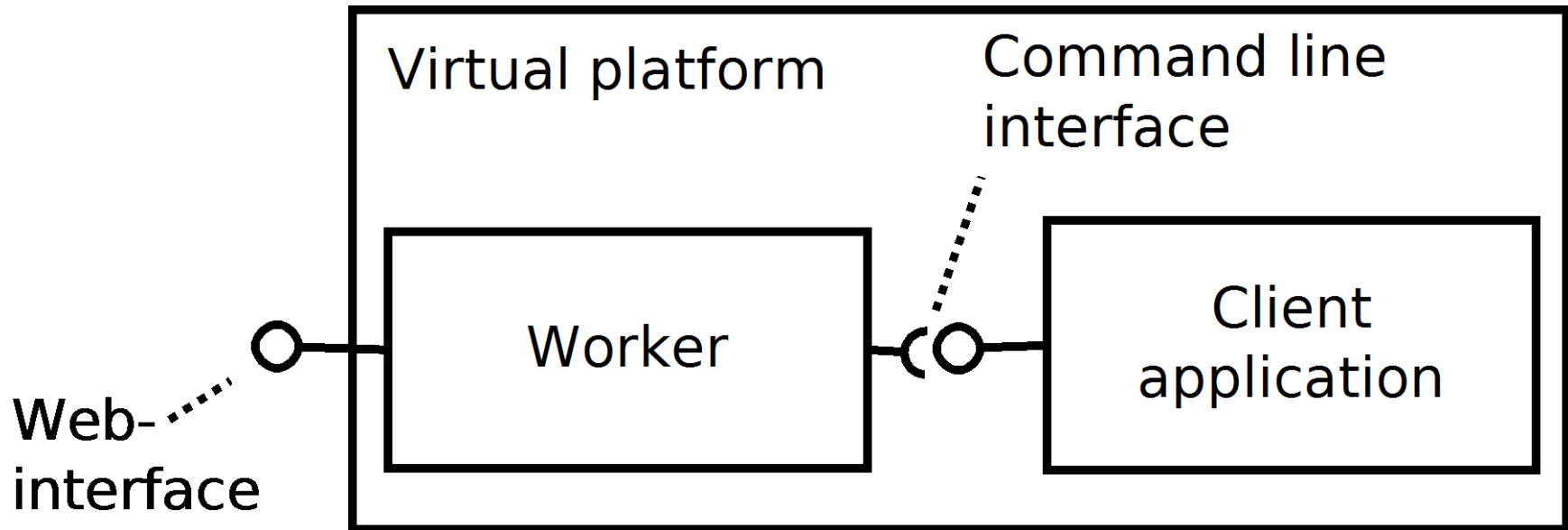


The screenshot shows the OpenShift console interface. The browser address bar displays https://openshift.redhat.com/app/console/application_types. The page is titled "Instant App" and "xPaaS". It lists several application types, including Jenkins Server, JBoss Data Virtualization 6, JBoss Enterprise Application Platform 6, JBoss Business Process Management Suite, and JBoss Fuse 6.1. A large text overlay asks "How do I deploy my application to the OpenShift?".

How do I deploy **my** application to the OpenShift?

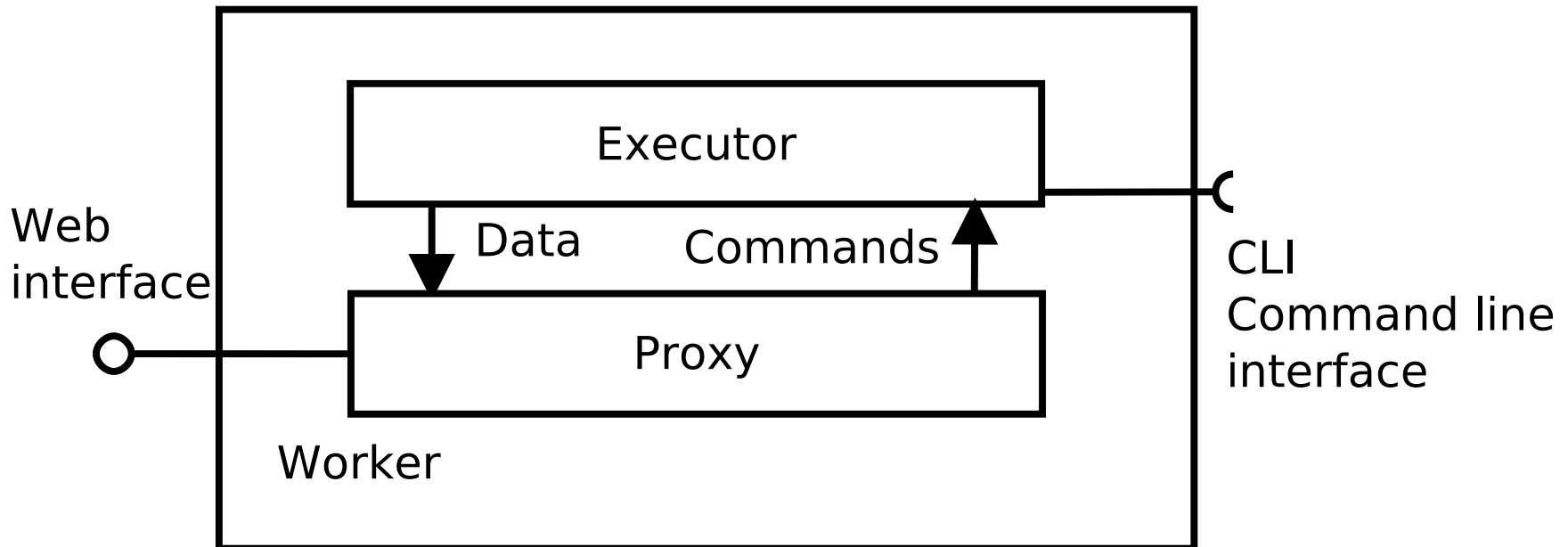
Mediation: Traditional Approach

- In order to support networking features without modifying an existing application a proxy component is required.



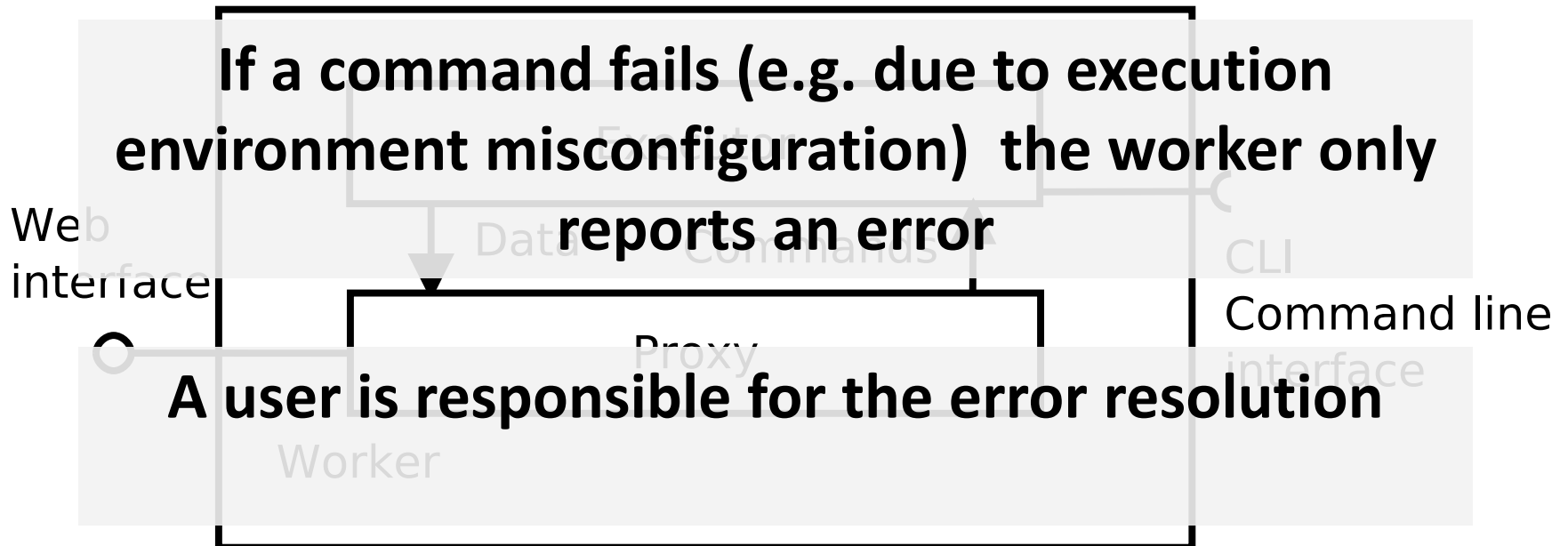
Mediation: Traditional Approach

- In order to support networking features without modifying an existing application a proxy component is required.

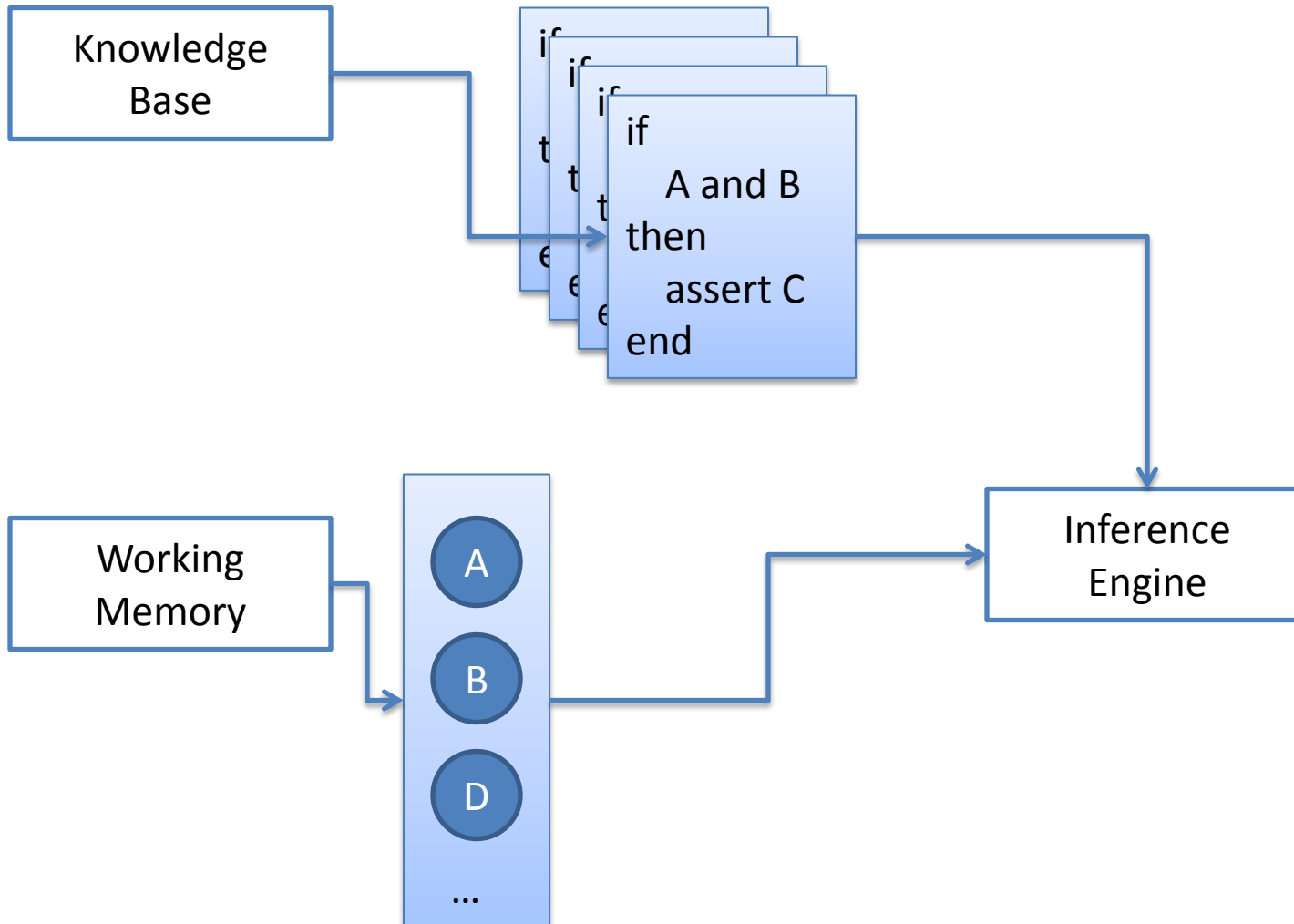


Mediation: Traditional Approach

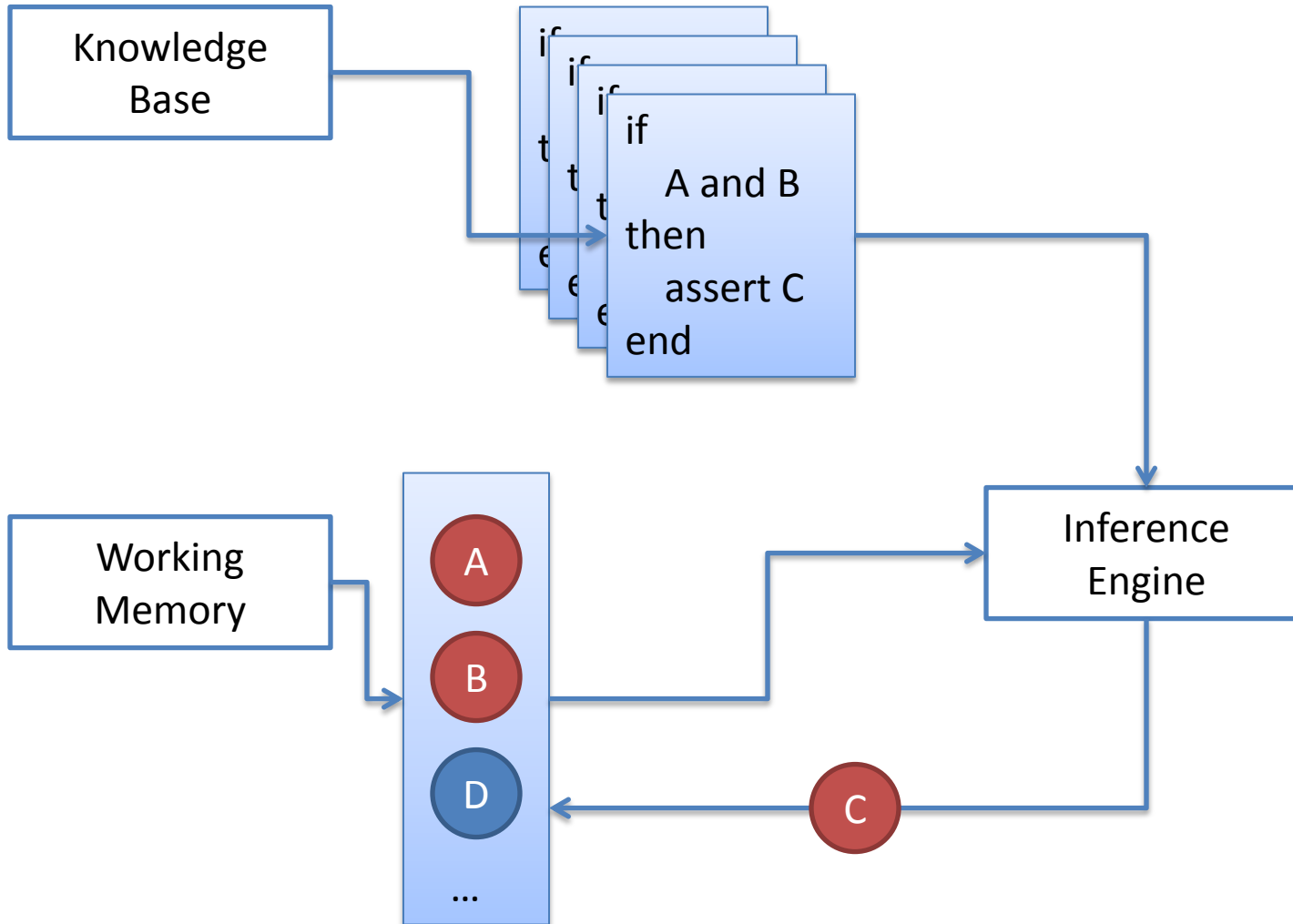
- In order to support networking features without modifying an existing application a proxy component is required.



Reminder: Expert Systems



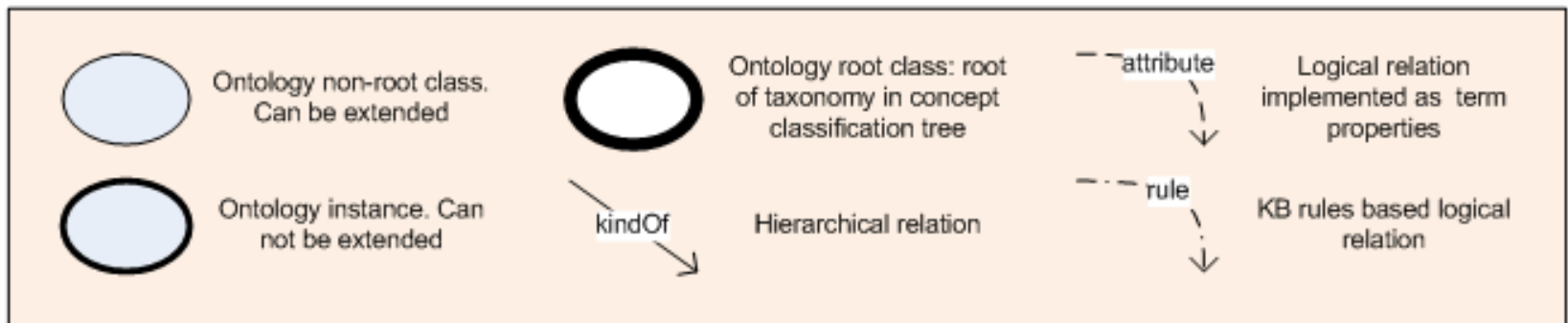
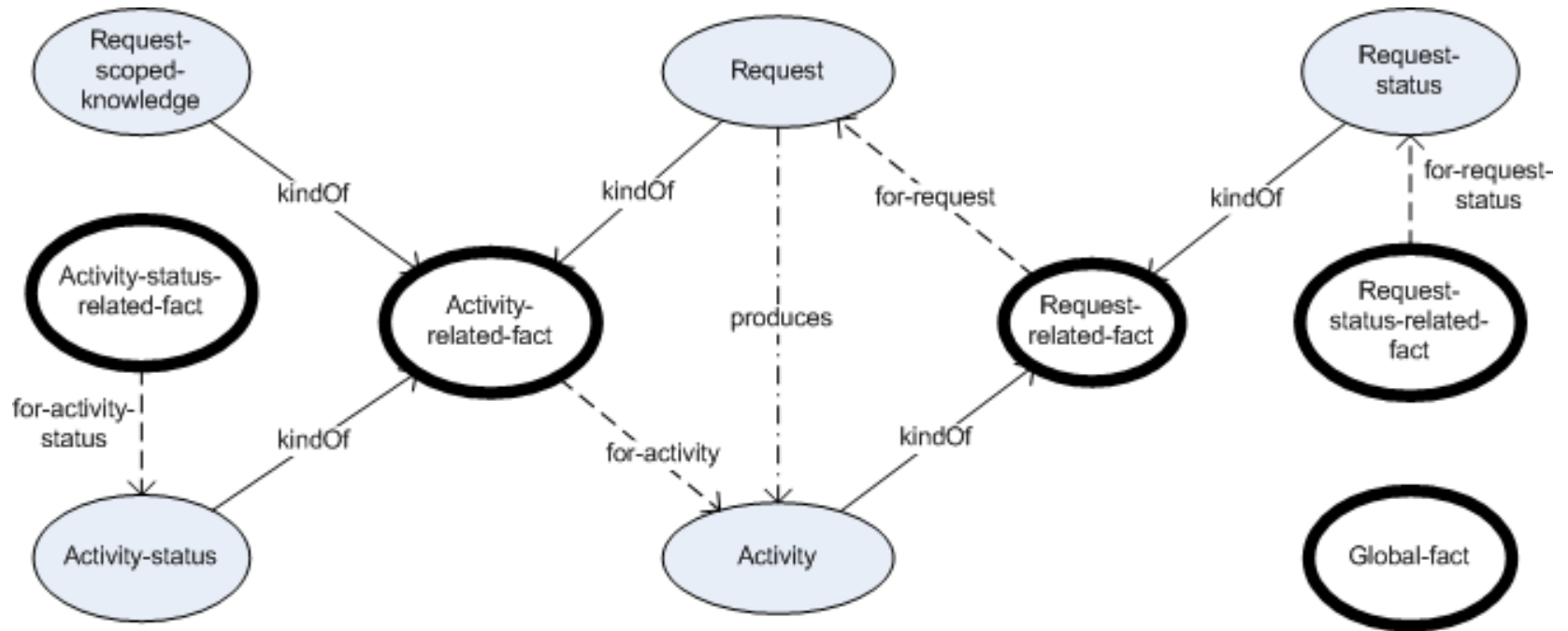
Reminder: Expert Systems



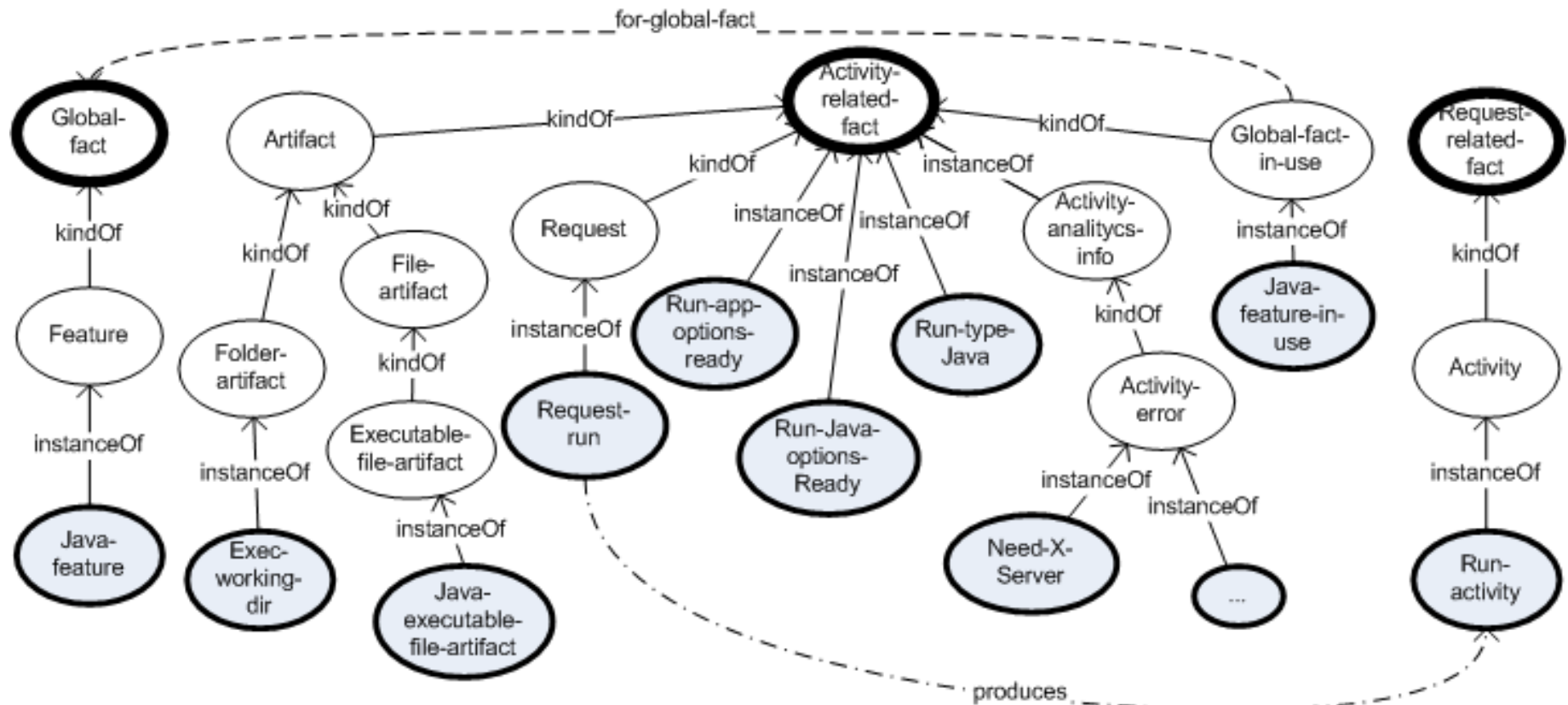
Software provisioning ontology as a core knowledge formalism ...

- ... to describe processes of software code building and execution
- ... to represent build and execution errors as well as actions required for fixing recognized errors
- ... to demonstrate how ontologies of specific tasks can be defined by extending the core ontology

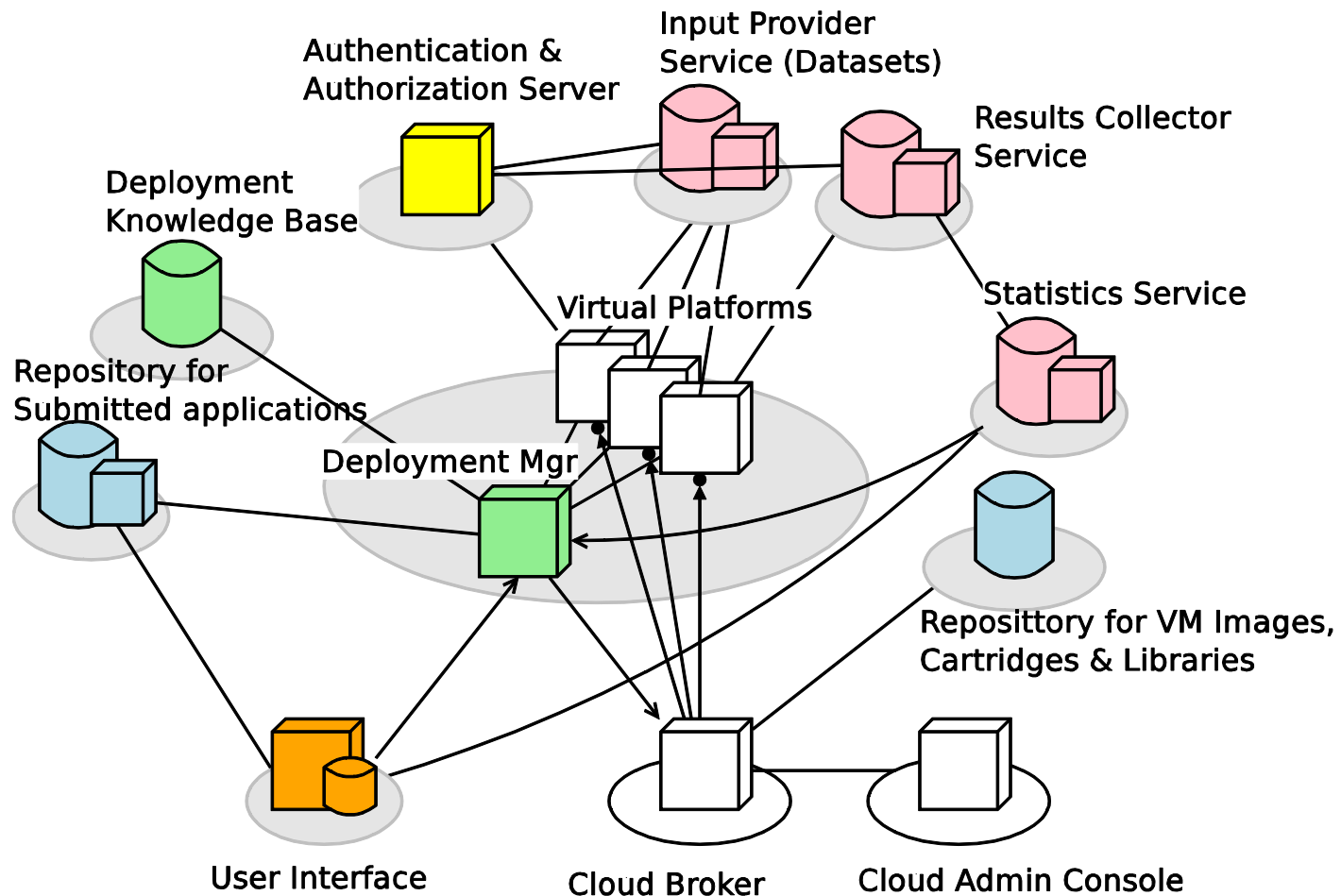
Activities and requests



Example of an ontology of specific tasks (Java)

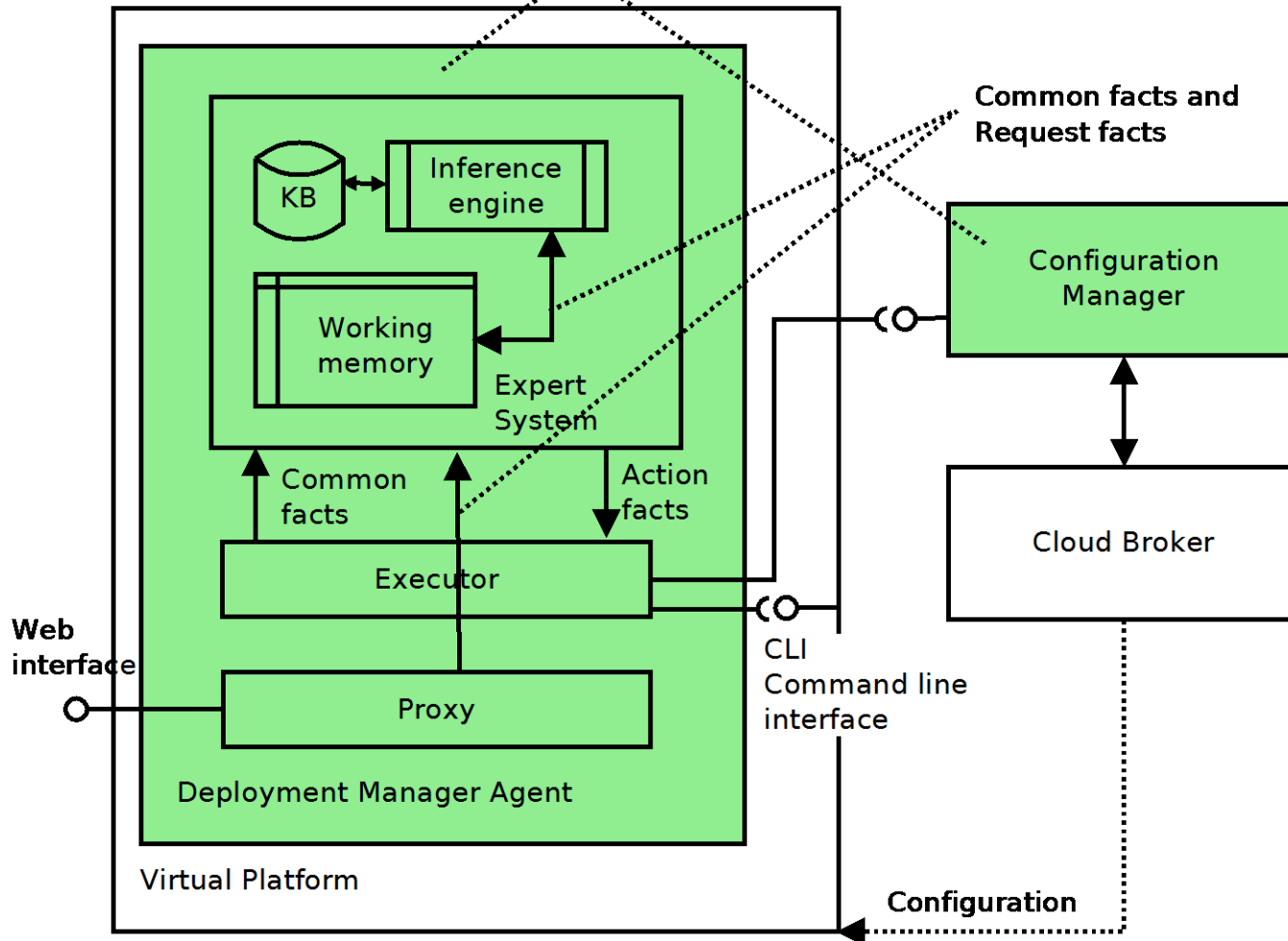


Software Provisioning Self-Service Networked Infrastructure



Deployment Manager Architecture

Deployment Manager: = Deployment Manager Agent + Configuration Manager



Evaluation

- We developed a system implementing proposed architecture as an **OpenShift cartridge**
 - Private cloud so far. Going public soon
- We successfully deployed (so far) 3 CLI applications in automatic mode
 - The most challenging part is knowledge representation

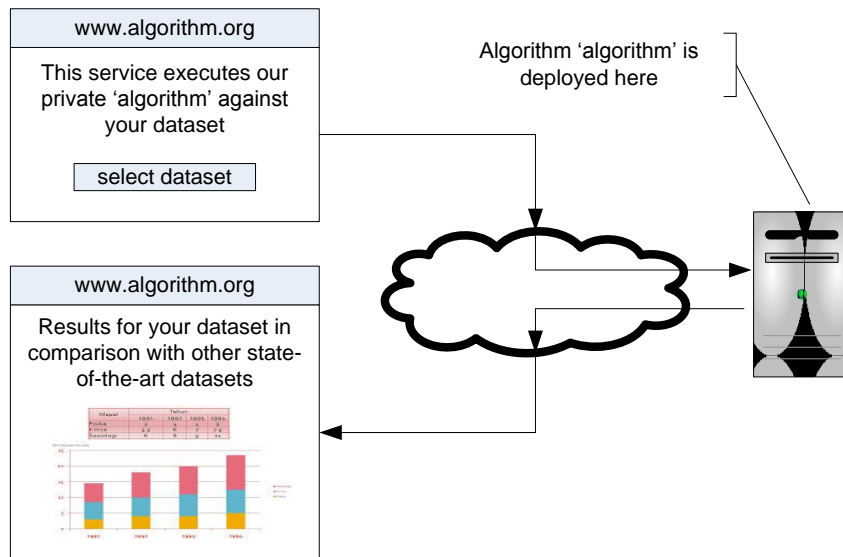
Summary

(architecture capabilities)

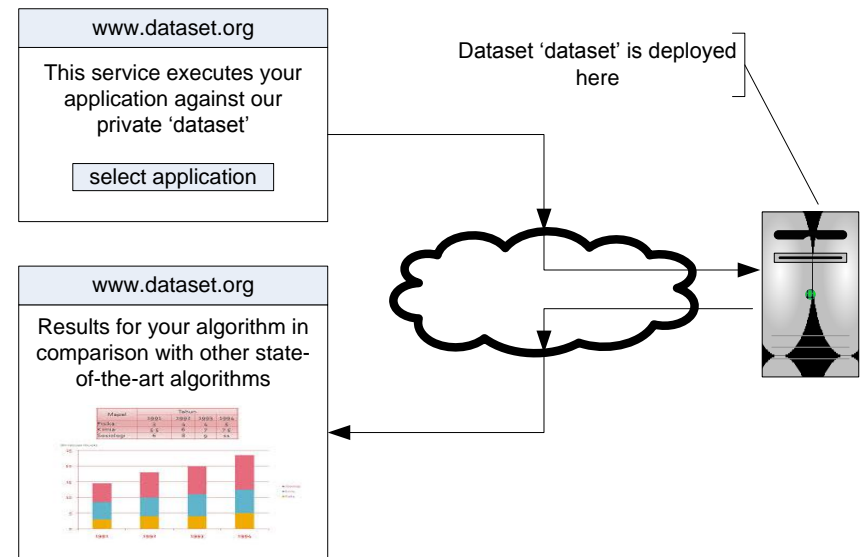
- **Able to learn:** Any technical issue needs to be resolved only once by an expert
 - Solution added to the knowledge base
- **Extensible:** New domain-specific tasks can be solved by only modifying the knowledge base
- **Platform independent:** Can be implemented to work in a cloud or in a local environment on any operation system. “CLI” can be changed to “Unified Interface”

Cloud in Theory

Algorithm as a Service



Dataset as a Service



Provisioning Service in Practice

Algorithm as a Service

Dataset as a Service

The screenshot shows a web browser window with the following elements:

- Browser Tabs:** jbossas | Open, HPCC Conferer, conference.hp, Automatic Depl.
- Address Bar:** cli-deploy.openshift.local:8080/kb/deploy
- Page Title:** Automatic Deployment Service for CLI Apps
- Text:** Please provide your application as a URL to any repository or as a single file and we'll do our best to deploy it to the cloud.
- Buttons:** Choose File (No file chosen), Upload file, Submit URI

On the left side of the image, there are two partial screenshots from another application:

- The top one shows a text area with "This service execu private 'algorithm' your datase" and a "select datas" button.
- The bottom one shows "Results for your da comparison with oth of-the-art data" and a bar chart with three bars.

On the right side, there is a diagram showing a server icon with a green dot and a red cross. A bracket labeled "deployed" points to the server icon. A large curly brace on the left of the server icon indicates a range or selection.



JYVÄSKYLÄN YLIOPISTO
UNIVERSITY OF JYVÄSKYLÄ

САНКТ-ПЕТЕРБУРГСКИЙ
ГОСУДАРСТВЕННЫЙ
ПОЛИТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ



Science 2.0 & Clouds for Better Research Collaboration: A Case of MIR

Q & (may be) A