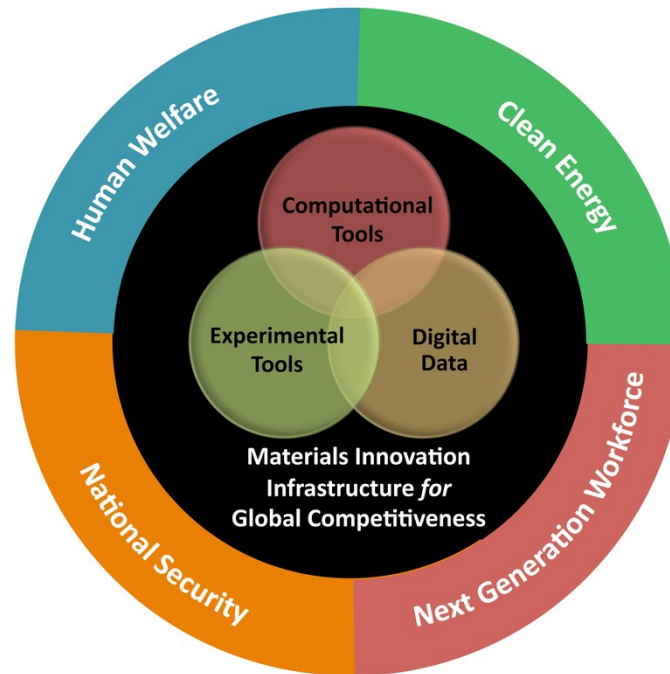


# A Cloud Service to Record Simulation Metadata



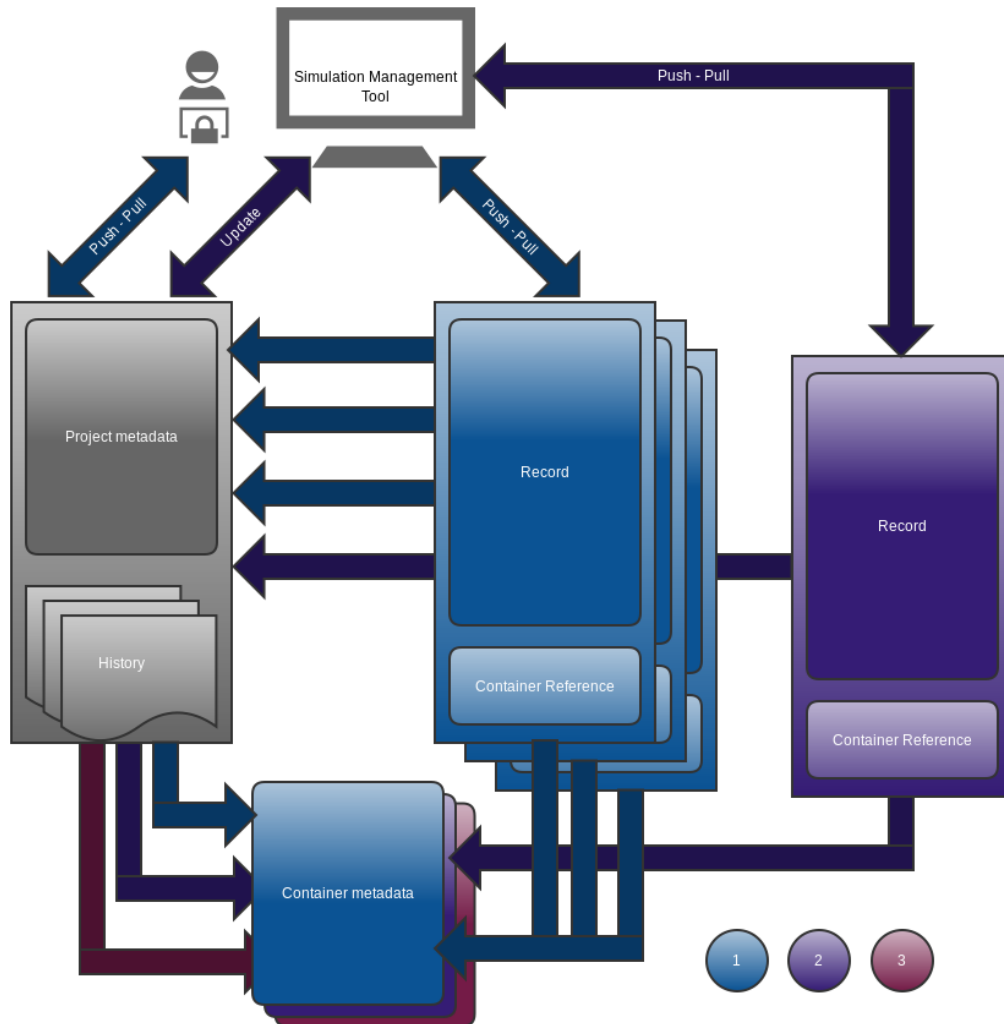
# Reproducible Research

- **Terminology:**
  - Reproducible
  - Repeatable
  - Replicable
- **Recording requirements & approaches:**
  - Workflows
  - Libraries
  - Event Control

# Motivation

- **Reproducible Record Atom**
  - Record metadata is not enough.
  - Not easy to go from metadata to a repeat.
  - Not always easy to get all the metadata.
  - Record Atom: Easy repeats and reproductions.
- **Cloud Service for record atoms**
  - Record metadata need easy sharing.
  - Cloud arena for research collaborations.

# Project-Container-Record



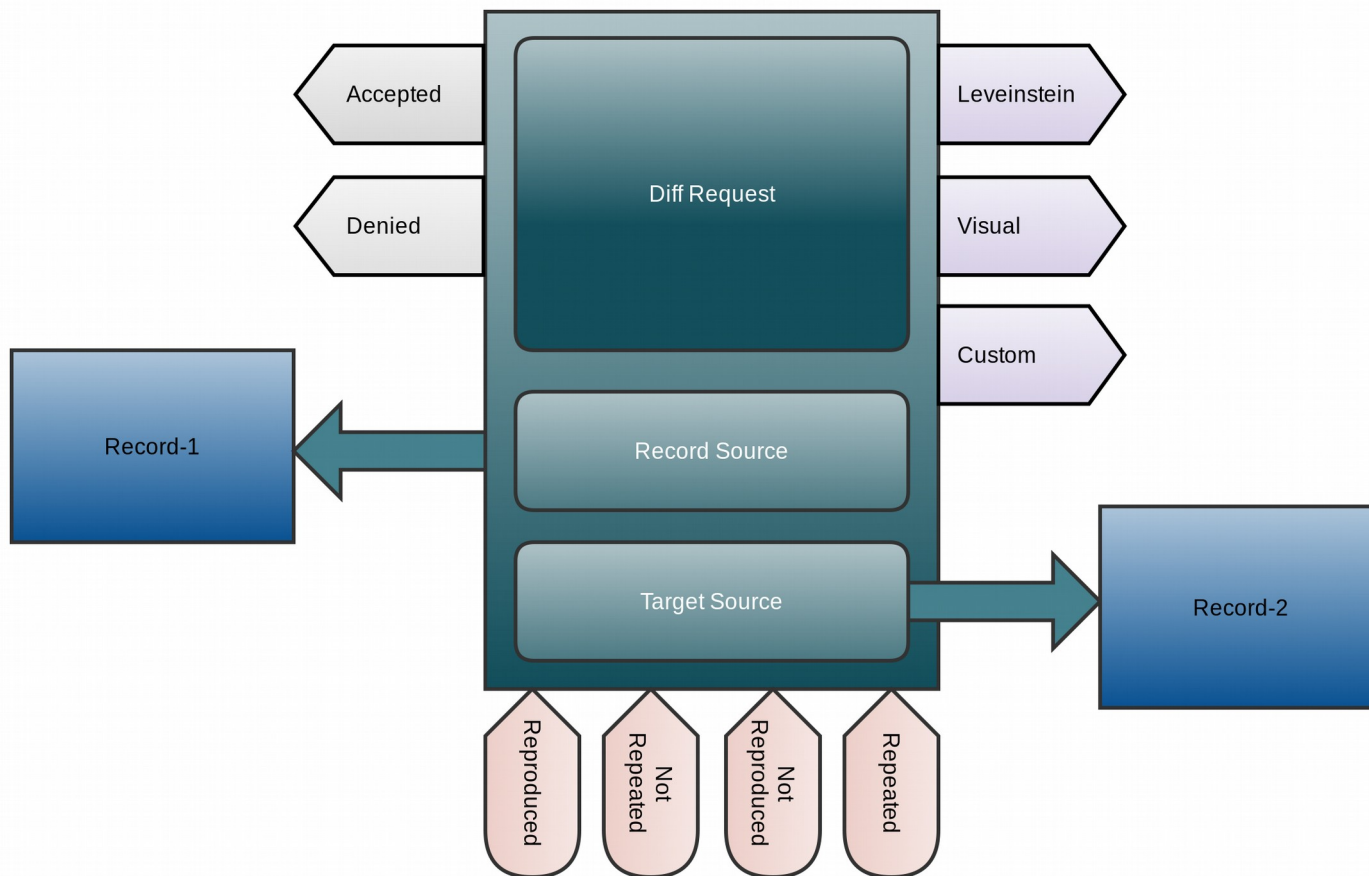
- **Project Management**

- Metadata push on creation.
- Versioning through history of containers per each major updates on sync.
- Relation one to many with the Record and Container.

- **Record Management**

- Meta data push on creation.
- Related to a project and a container in the history.
- Relation many to one with Project and Container.

# Collaboration Entity



- **Record and Project Reproducibility assessment**
- **Cloud interaction on Project from record atoms**

- **Status:**

- Accepted
- Denied

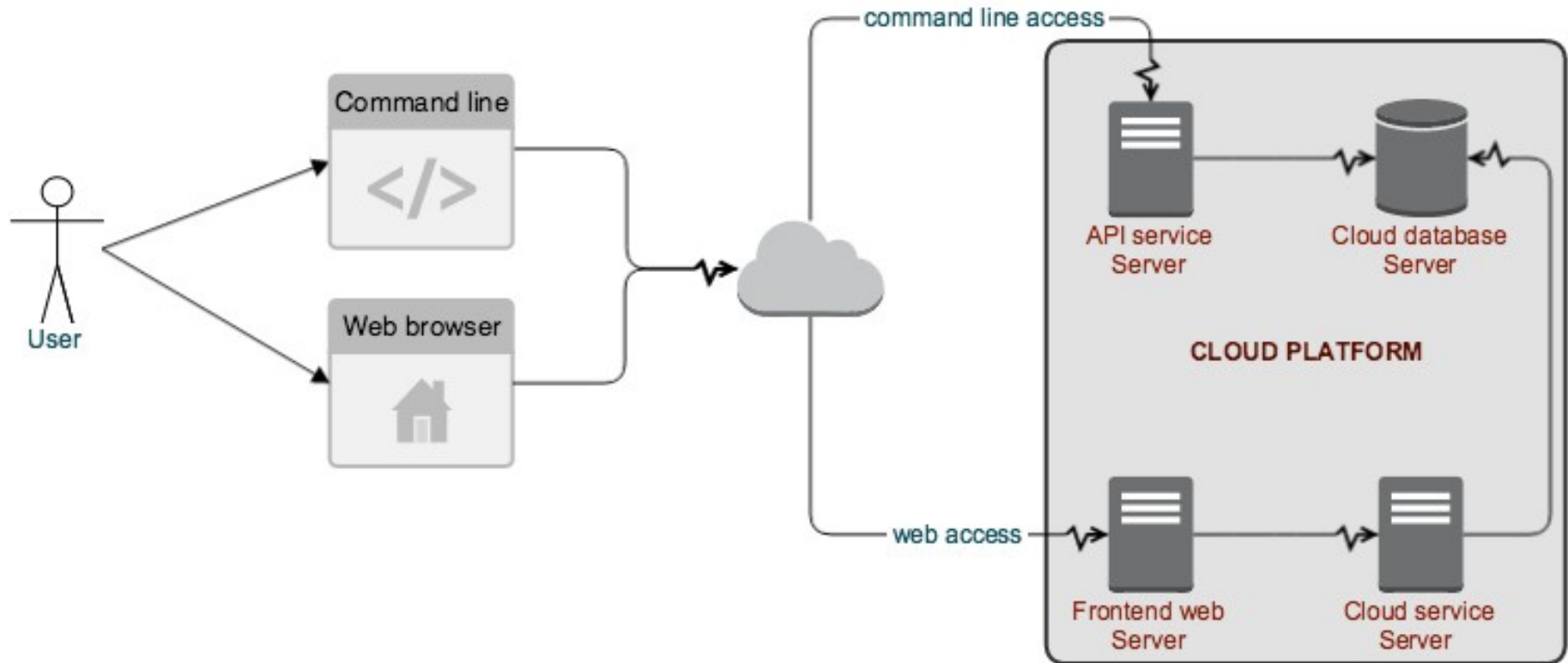
- **Method:**

- Leveinstein
- Visual
- Custom

- **Assessment:**

- Repeated
- Reproduced
- Not Repeated
- Not Reproduced

# Reproducible Record Atom Cloud Service



- **Record and Project Reproducibility assessment**
- **Cloud interaction on Project from record atoms**

# Usage Requirements

- **Simulation Management Tool**

- Implement the API access to the platform.
- Container generation and push during runs.
- On the fly updates on execution state.

- **Scientist Simulation**

- Create an account and get API key.
- Wrap simulation in Container.
- Use a supported simulation management tool.

# Platform Status and Technologies



Apache  
Tomcat



# Flask

web development,  
one drop at a time



mongoDB



docker

*Sumatra*



# Project Instrumentation Demo

- **Step1: Create a user account and login.**
- **Step2: Retrieve the API key and check dashboard.**
- **Step3: Clone the github ddsd-demo code.**
- **Step4: Update Dockerfile.**
- **Step5: Update the manage script.**
- **Step6: Build the first Project image.**
- **Step7: Run the simulation the first time.**
- **Step8: Check personal online dashboard.**
- **Step9: Run the simulation a second time.**
- **Step10: Pull a record.**

# Perspectives

- **Command line executer from a pull**
- **Frontend improvement**
- **Platform features:**
  - Automatic container image generation from Record metadata.
  - On the cloud execution of a record atom capability with Kubernetes.
  - Reproducible Project Record Atom Identification: RPRAI
- **More container systems:**

Rocket
- **More simulation Management systems:**

workflow, library and event control based.

# Acknowledgement

- **PhD supervisors:**
  - Dr. David Hill
  - Dr. Jonathan Guyer
- **NIST colleagues:**
  - Dr. Andrew Reid
  - Dr. Stephen Langer
- **Early collaborator:**
  - Dr. Daniel Wheeler

# Research-Implement-Run&Share-Collaborate

Thank you

- **Demo:**
  - **Source:** <https://github.com/faical-yannick-congo/ddsm-demo>
  - **Data:** <https://github.com/faical-yannick-congo/ddsm-demo-data>
- **Presentation:**
  - **slides:** <https://github.com/faical-yannick-congo/scipy2015-talk>