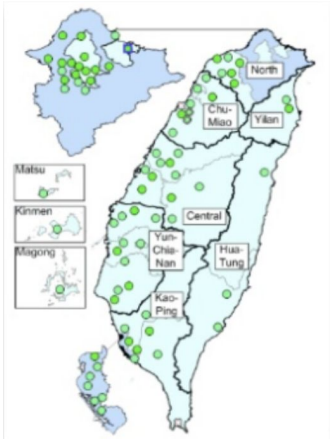
SEADTrain and E-RPID Environmental Sensor Use Case

RDA GEDE Webworkshop - 22 May 2019

Rob Quick, Marina Krenz, and Yu Luo

SEAD and SEADTrain

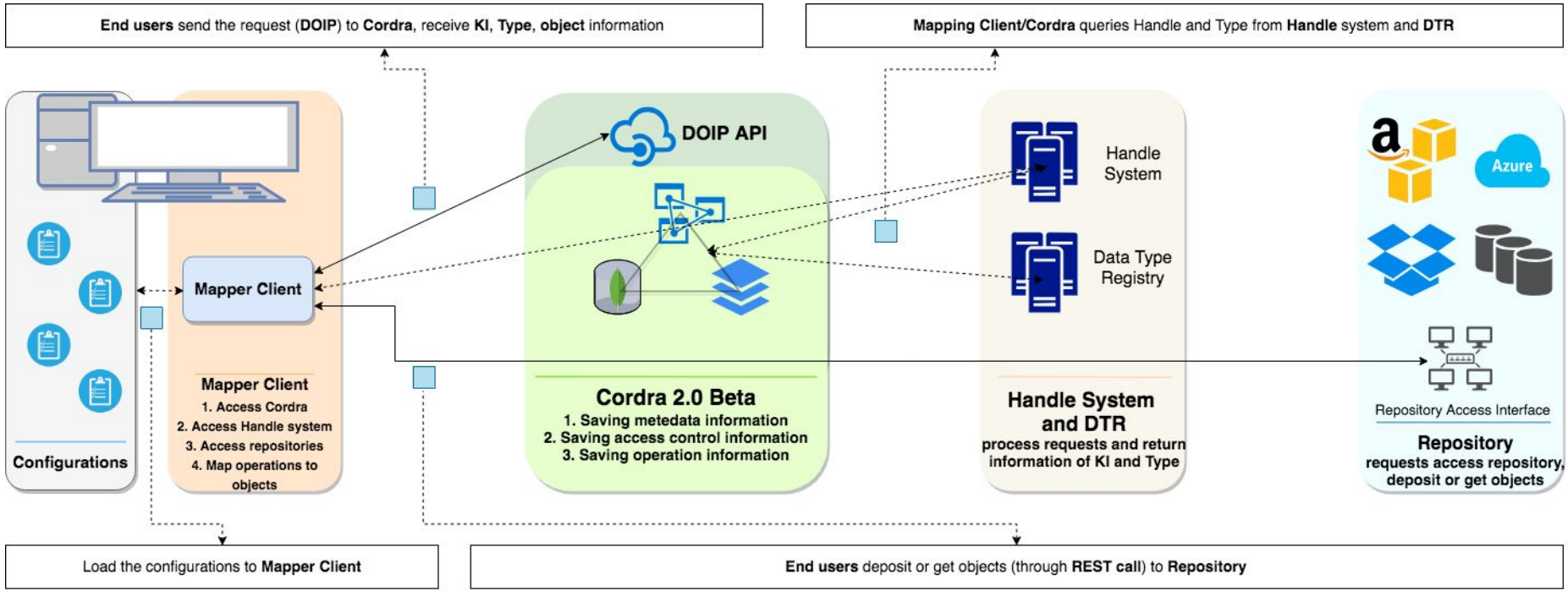


- Sustainable Environments/Actionable Data
  - http://sead-data.net/
  - Data hosted on US National Data Services
  - Variety of environmental data
- SEADTrain
  - https://github.com/Data-to-Insight-Center/SEADTrain
  - Provides a hands-on, project-based experience for the on-line student of data science.
  - Initial data set used for this research includes AirBox sensors throughout the island of Taiwan.
  - During training events data was homed on Azure.
  - Web interface using RPID services: https://data-to-insight-center.github.io/SEADTrain/
  - Allowed high level filtering based on RDA PID Kernel Information strawman profile. http://bit.ly/2oH53XC

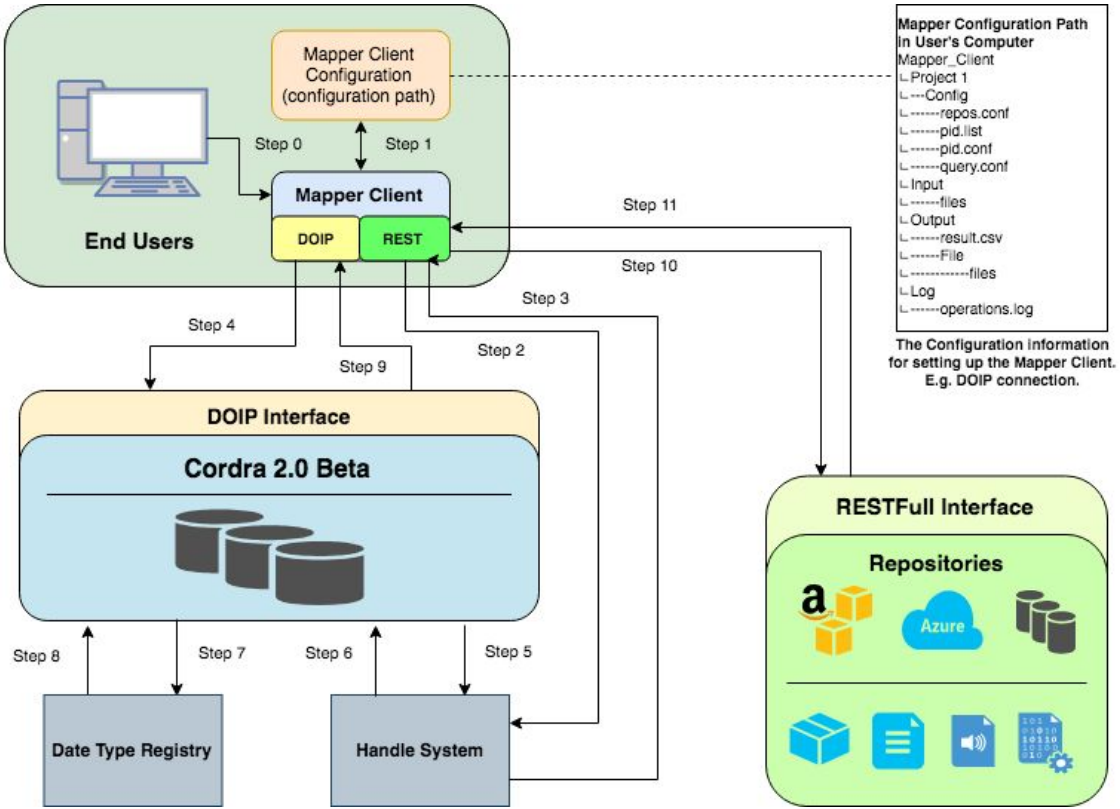
The RPID/E-RPID Testbed Projects

- RPID Project (NSF Grant 1659310) - https://rpidproject.github.io/rpid/
  - $200K - March 1, 2017 to February 28, 2018 (+1 yr NCE)
  - Testbed combining Handle Service, Data Type Registry, 4 Diverse Use Cases, PID KI, and User Advisory Board.
  - Puppetized installations. Services on AWS and bare metal.
  - Much of the foundational setup needed to move to the next steps of research.
- E-RPID
  - $300K - October 1, 2018 to September 30, 2020
  - Proposal to research DOIP and services necessary to utilize it.
  - Repository mapping, DOIP, 2 existing and 2 new use cases, training materials.
  - Moved to JetStream cloud using Openstack.
  - Still early in this award so mapping services, DOIP research, and use case integration are in early stages.

Architecture (Diagram)



Workflow (Diagram)



Step 0: A user runs the Mapper Client with a PID and an operation

Step 1: Load the configurations into Mapper Client

Step 2: the user sends the PID of a object to Handle system

Step 3: Handle system returns the PID Kernel Information

Step 4: Client parses PID KI and accesses the Cordra object with a DOIP request

Step 5: Cordra accesses the Handle object for extra information

Step 6: Handle system return extra information of Cordra object

Step 7: Cordra accesses the DTR for Type information

Step 8: DTR returns the Type definition for the Cordra object

Step 9: Cordra returns the access control information and metadata information

Step 10: Mapper Client sends a REST call to the Repository

Step 11: The Repository returns the file or other requested data

Core Component: **Mapper Client**
It sets up a service based on the local configuration files, sends requests to Cordra (**DOIP**) and Handle system (**REST**) to acquire the PID KI, metadata and access control information, maps the operation to the data object, and finally accesses the object within the Repository.

Challenges

1. Providing proper mapping between requirements and digital objects while using Digital Object Architecture.
2. Registering persistent identifiers (PIDs) for objects within Repositories, and ensure the FAIR (Findable, Accessible, Interoperable and Reusable) principles in digital objects.
3. Utilizing the DOIP (Digital Object Interface Protocol) to access the digital objects
    a. Providing sufficient operations
    b. Objects are properly assessed by DOIP based on users' requirements
4. Access control and Security

Effort and Future Directions

- Effort
  - RPID - 0.25 FTE PhD Student and 0.1 FTE PI
  - E-RPID - 0.25 FTE PhD Student, 0.1 FTE Developer, 0.1 FTE PI
- E-RPID testbed are experimenting with
  - Mapping and brokering processes to create a virtual object level over existing information structure
  - Using DOIP to optimize interaction with the object layer by understanding how to query each object for the set of operations that apply to it
- Future Directions
  - Stable and flexible mapping services to create the virtual object level by using DOIP, PID and Digital Object Architecture
  - Contributing to FAIR within Open Science