ViCE Registry

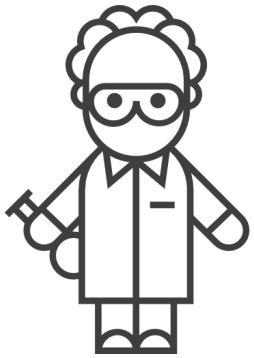
An Image Registry for Virtual Collaborative Environments

Institute of Information Resource Management
University of Ulm

Christopher B. Hauser, **Jörg Domaschka**

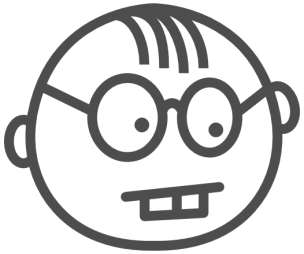
CloudCom 2017
Hong Kong

Virtual Environments in Modern Computing



Chemical Scientist describes computation and submits them to HPC cluster with MOAB

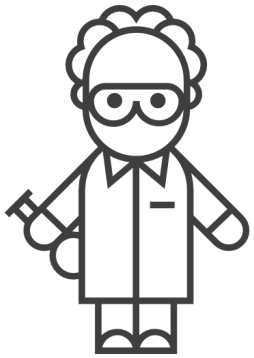
Lecturer prepares desktop PCs for student exercises with specific software and configuration



Software developer writes Dockerfiles to encapsulate the software application as a container

System administrator prepares virtual/physical machines with software to serve required services

Virtual Environments in Modern Computing

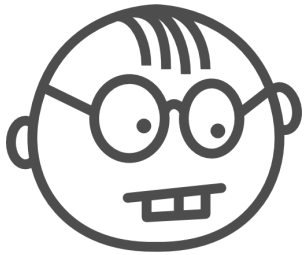


Virtual Environment

Execution Environment

Virtual Environment

Execution Environment



Virtual Environment

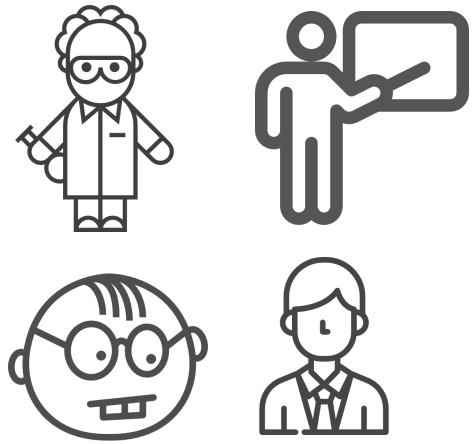
Execution Environment

Virtual Environment

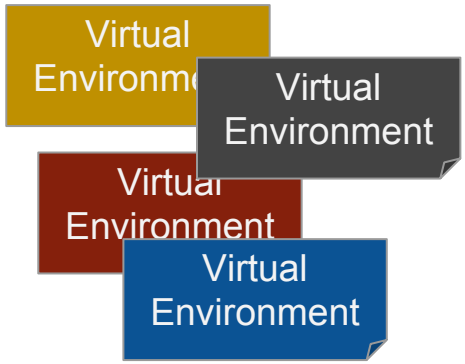
Execution Environment

Diversity in Modern Computing

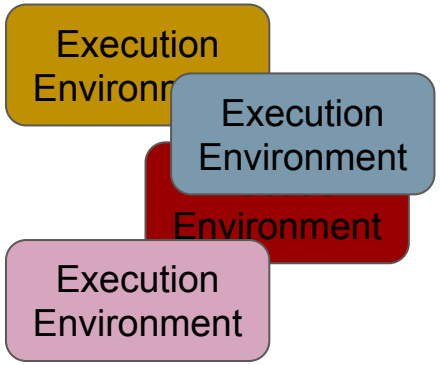
Communities



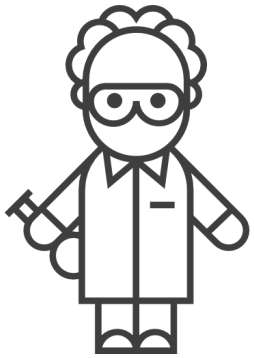
Definition



Execution

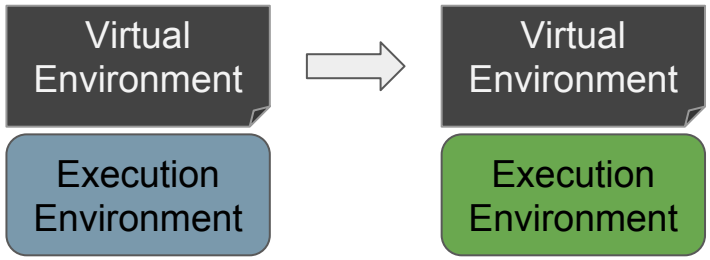


Use Case Examples



Describe HPC job to run in the future, and on other execution environment

Prepare desktops for students to work on PCs in class, but also in the cloud from home

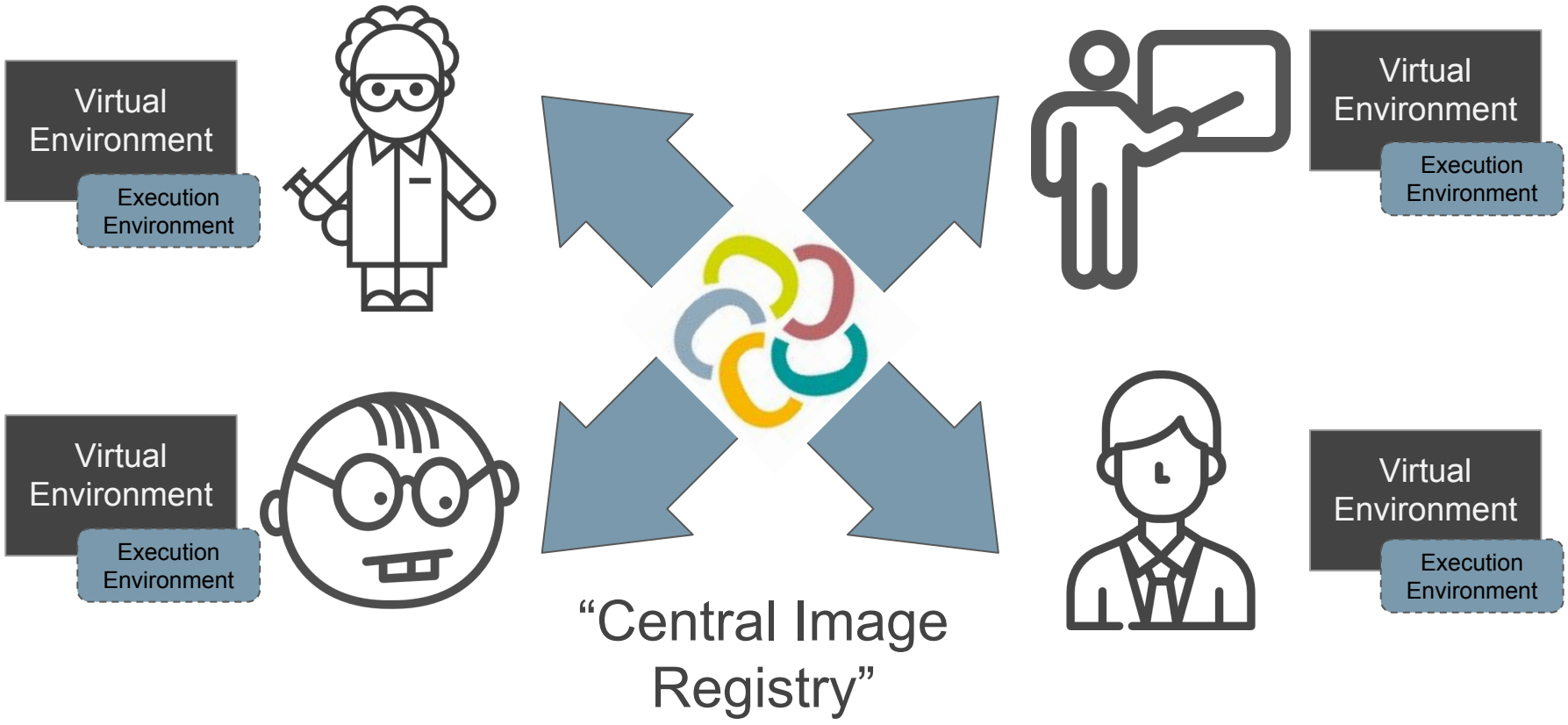


Problem Statement

- Possible solutions?
  - Standardized Specifications (OVF, OCI)
  - Common Execution Middleware (e.g. Java VM)
  - Use Repositories (OpenStack Glance, Docker Hub)

- Why another approach?
  - Need for independent classification of
    - execution environments
    - virtual environment definition
  - Extendable for new environments

A Central Image Registry



Motivation & Problem Statement

=> Central Image Registry for Virtual Environment

=> ViCE Registry bridges the gaps between:

Organizations

Execution Environments

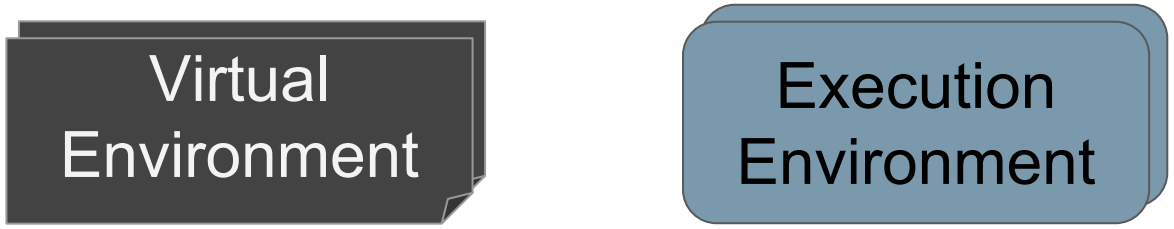
Communities

ViCE Registry

Concept & Contribution

Contributions of ViCE Registry

1. Classification: Definition & Representation

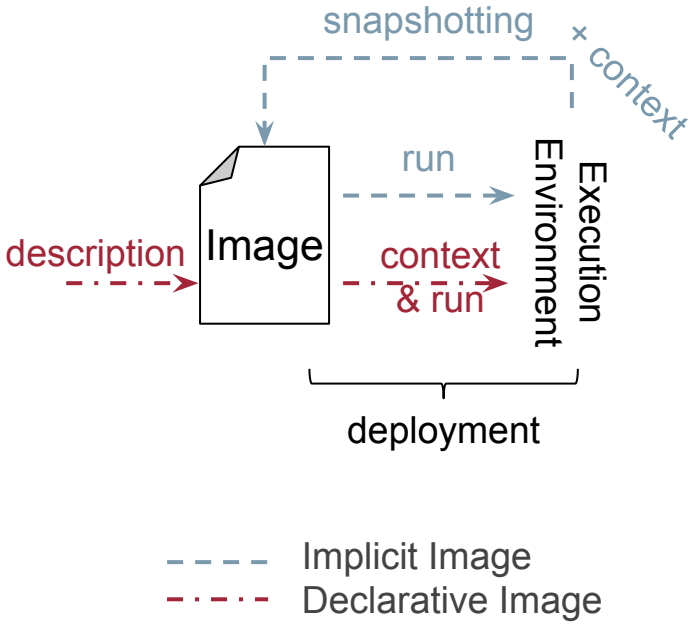


2. Registry Architecture and Implementation

Classification of Virtual Environments

Virtual Environments, defined by

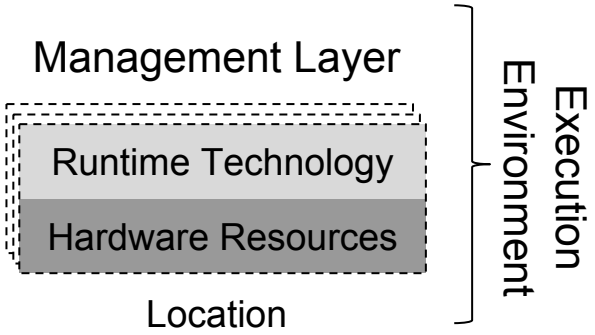
- Image type
  - declarative / implicit
- Content type
  - image file type
  - e.g. qcow2, Dockerfile, Moab job description
- Dependency to Execution Environment



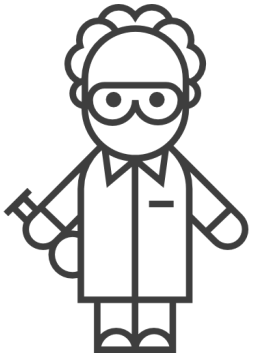
Classification of Execution Environments

Execution Environments, defined by

- Hardware
  - CPU architecture & model, Memory, Storage, ...
  - Infiniband interconnect
- Runtime Technology
  - “Bare metal”, Containers, Virtualisation
  - Software: e.g. KVM, Docker, …
- Management Layer
  - Basic, Cloud, Container Cluster, Job Scheduler
  - Software: e.g. OpenStack, Kubernetes, ...



Classification for Use Cases (1)



Describe HPC job to run in the future, and on other execution environment

Execution Environment	4 Servers, Infiniband networking
	Dual Socket Intel Xeon, 2.6GHz, 32 GB DDR3 RAM
	Centos 7 with Kernel 3.2 and kernel modules x,y,z
	MOAB job scheduler
Virtual Environment	Required Modules (m_1, m_2, m_3, …)
	Requested resources (cores, memory, time)
	Executable job description

Classification for Use Cases (2)

Prepare desktops for students to work on PCs in class, but also in the cloud from home

Execution Environment	Desktop PC with Intel i7 3.2GHz, 8GB DDR3
	1TB HDD, Gigabit Ethernet Connection
	Keyboard, Mouse, Monitor
Virtual Environment	Windows 10 Education
	Software Artefacts (Excel, Eclipse, …)
	Configurations

Detailed Classification Schema

TABLE I
GENERAL METADATA

field	description [content type]
image id	an unique identifier of an image [string]
version	version number of the image [increasing integer]
base image	reference to previous/base image [image id]
status flag	indicate the availability of the image [active, deleted]
creation	timestamp of image creation [date time]
update	timestamp of latest image update [date time]
owner id	reference to the image owner [integer]
group id	reference to the image group [integer]
permissions	access control for owner/group/others [rw,rw,rw]
title	image title, used for the catalogue entry [string]
description	image description in the catalogue entry [string]
community	target community, e.g. biochemistry [list of strings]
purpose	the usage field e.g. teaching, research, SaaS [list of strings]
total	number of exports in total [integer]
active	number of running deployments [integer]
ratings	a list of user ratings between 1 and 5 [list of integers 1-5]

TABLE II
IMAGE ORIGIN METADATA

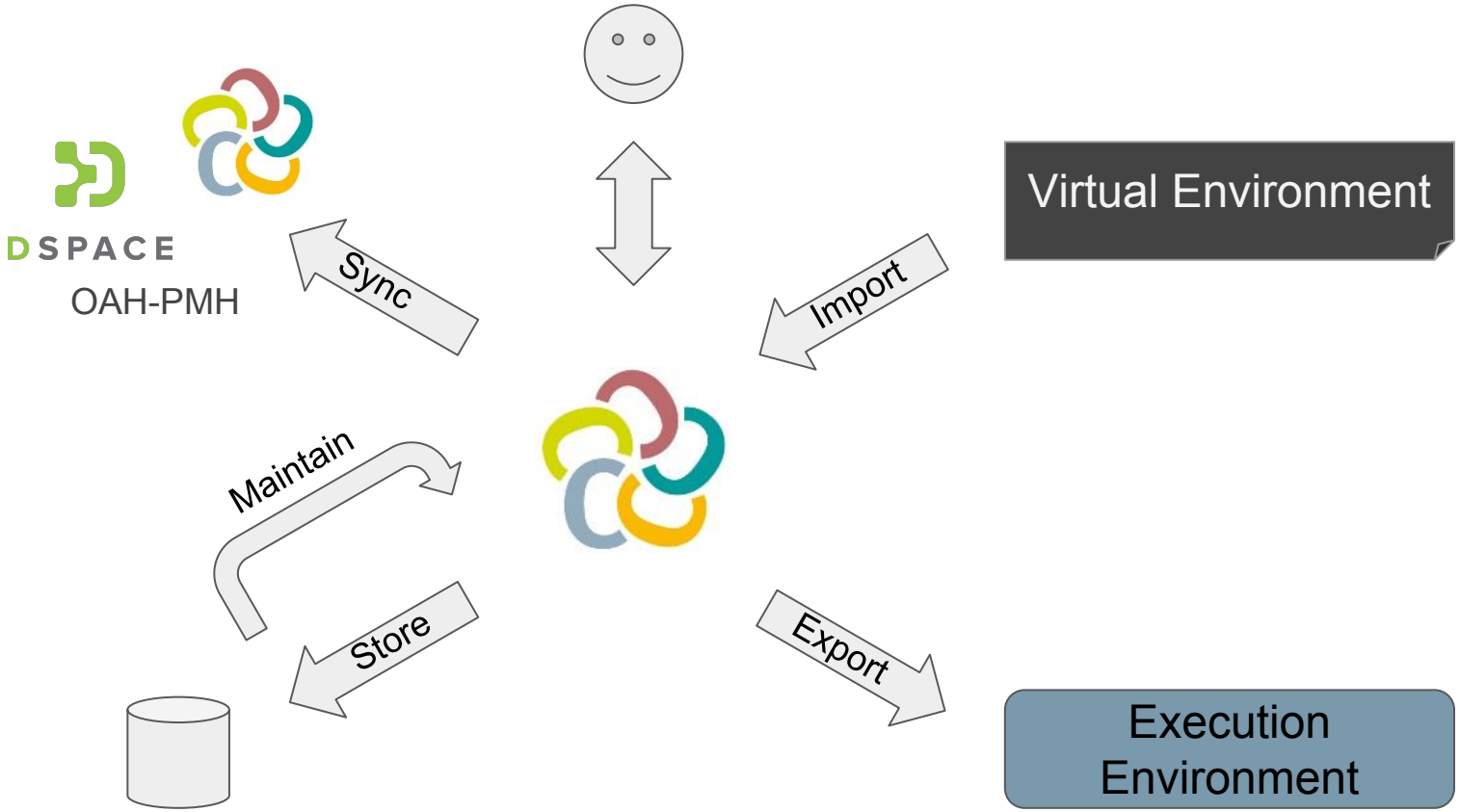
field	description [content type]
runtime technology	cf. IV, [vm, container, application]
runtime tech. impl.	software & version, [kvm 4.10.11, ...]
management layer	cf. IV, [basic, cloud, container, scheduler]
mgtm layer impl.	software and version, [openstack, kubernetes, ...]
location	ref. to origin location, [e.g. ref. to openstack setup]
image-type	cf. IV [implicit, declarative]
content-type	image file type [qcow2, Dockerfile, Packerfile, ...]
content-refs	list of additional requirements, e.g. files, images, ...
checksum	checksum to validate image correctness [string]
virt. hardware	virtual hardware (vCores, storage, memory, ...)
env. hardware	physical hardware (CPU arch., storage type, ...)
resource profile	typical resource utilisation (CPU or I/O usage, ...)
minimal hardware	a virtual hardware profile to deploy this image

TABLE III
SEMANTIC METADATA

field	description [content type]
execution type	[service, job, interactive]
user interaction	interaction to image instances [direct, indirect]
user interface	main access virtual environment [cli, gui, web, none]
operating system	e.g. Ubuntu 14.04 LTS Server, ... [list of strings]
software packages	list of software with version [list of strings]
service depends.	required services e.g. licensing, authentication, ...

Note: please find detailed descriptions in the Paper.

Registry Tasks and Requirements



OAI-PMH: Open Archives Initiative Protocol for Metadata Harvesting

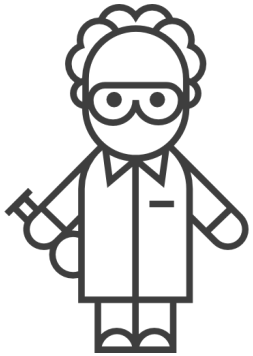
Use Case Examples

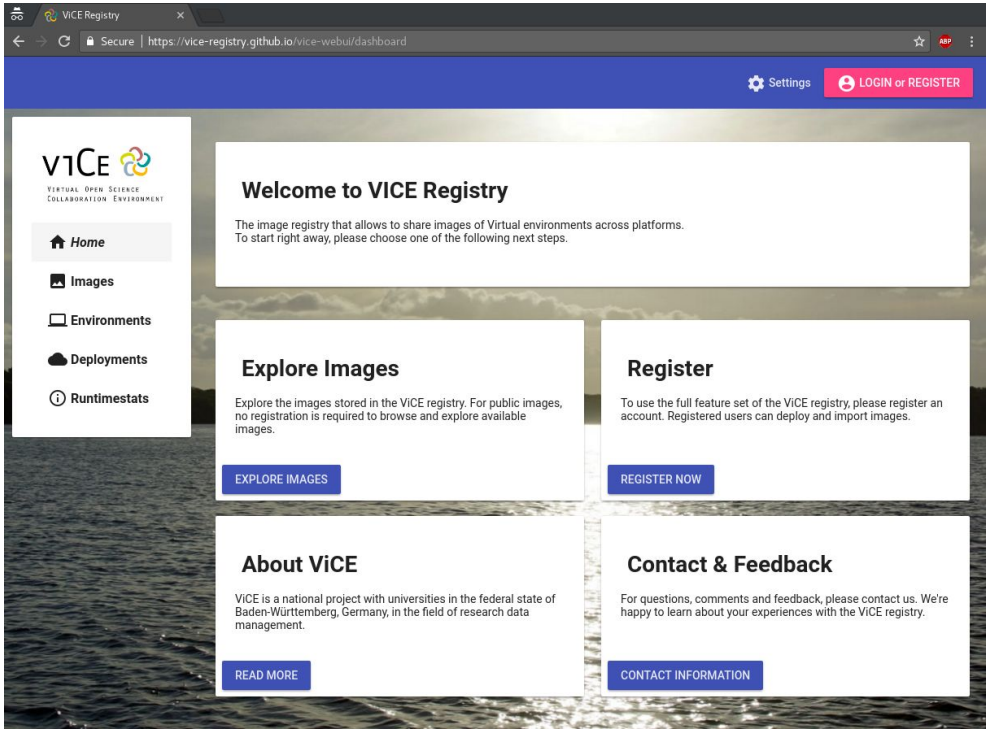
- Scientist can move jobs from cluster a to b
- Scientist can archiv his jobs and run later
- Lecturer can provide Images for IaaS Clouds
- Admin can roll out his environment across multiple clouds

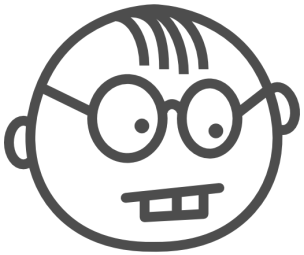
ViCE Registry Implementation

Architecture & Evaluation

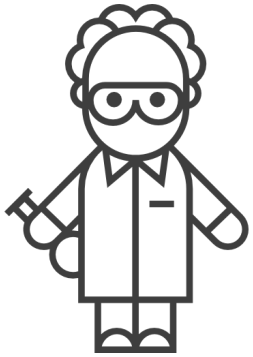
Graphical User Interface

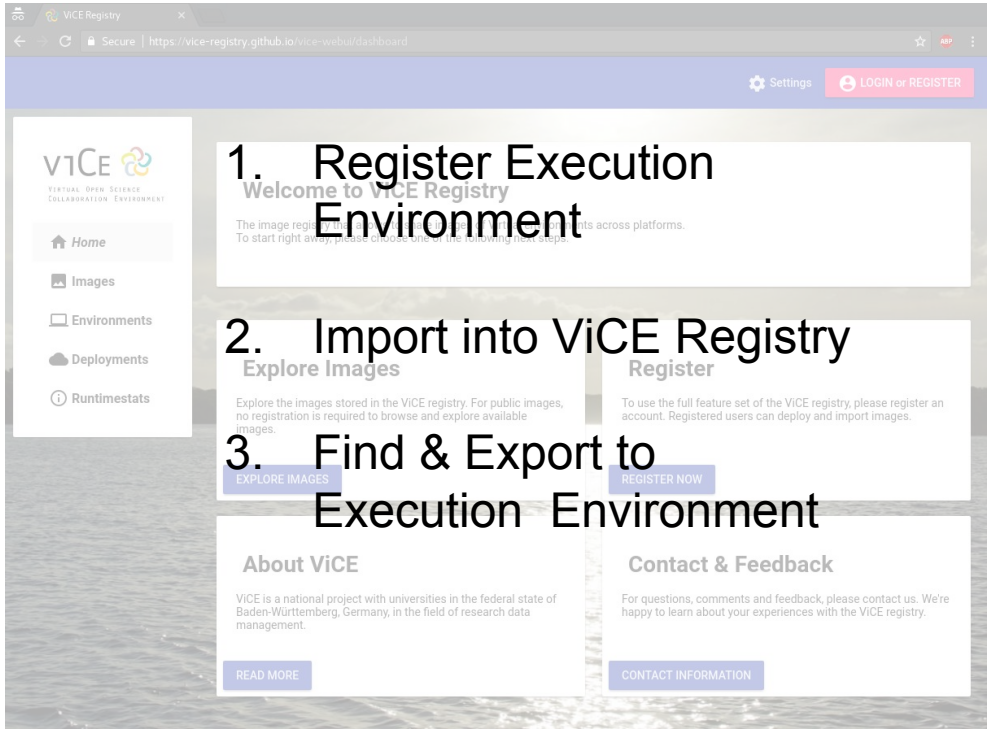




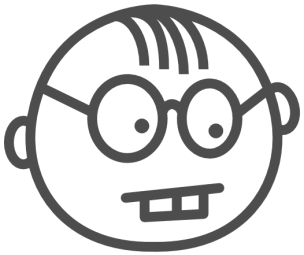


Graphical User Interface

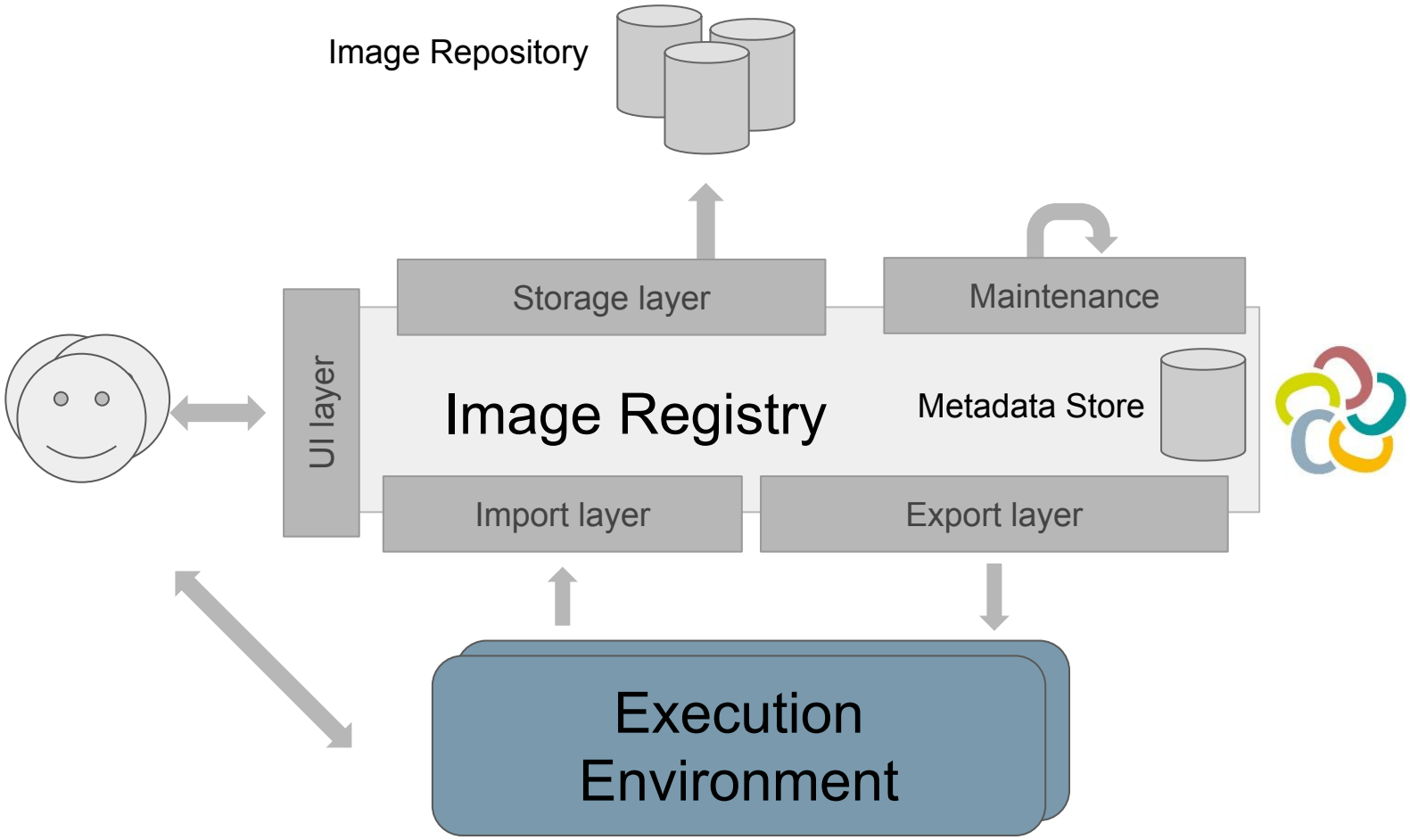




1. Register Execution Environment
2. Import into ViCE Registry
3. Find & Export to Execution Environment



ViCE Registry Architecture



Conclusion

Evaluation, Summary & Outlook

Feature Evaluation

- ★ Import & Export of Images
- ★ Support multiple execution environments
- ★ Framework with basic features

  ⇒ Still under development

TABLE IV
VICE REGISTRY CROSS PLATFORM SUPPORT

	OpenStack	bwLehrpool	Docker	Singularity
OpenStack	✓	✓	✗	✗
bwLehrpool	✓	✓	✗	✗
Docker	✓	✓	✓	✓
Singularity	✓	✓	✓	✓

✓fully supported, ✗not yet supported

Implementation & Deployment

- Written in Go
- Web UI with Angular 2

- API Definition with Swagger
- Microservice Architecture

Open Source on Github
https://github.com/vice-registry

Shipped with Docker
https://hub.docker.com/u/viceregistry/

- Simple deployment with `docker-compose up`, cf. https://github.com/vice-registry/vice-registry

Summary

- Virtual Environments, described as Images
- Execution Environments of different types

- Central Image Registry as exchange point

- Contributions:
  1. Classifications
  2. Implementation (ongoing work)

Outlook

- Finish Implementation
  - Add support for more execution environments
  - Add maintenance layer (“task runner”)
  - Launch a prototype for Universities in Baden-Württemberg
- Extend Classification
  - e.g. add resource usage statistics to images

ulm university universität ulm

Questions and Comments?

Institute of Information Resource Management
University of Ulm
christopher.hauser@uni-ulm.de

Registry Tasks and Requirements

- Import / Export Virtual Environments as Images
  - Declarative or Implicit
  - Convert Image as needed
- Store and maintain Images
  - archive images for later use
  - e.g. nightly updates & build
- Support (User) Interfaces for End Users and Synchronisations
  - Web User Interface
  - Open Archive Initiative Protocol for Metadata Harvesting (OAI-PMH)
- Support of various Infrastructures (Execution Environments)
  - OpenStack Private Cloud
  - Amazon EC2 Public Cloud
  - Docker Swarm / Kubernetes clusters
  - HPC Clusters with Singularity

ViCE Registry Components