

Historical Transition of Repository Functions and Latest Trends of Repository System Architecture

Masaharu Hayashi, Kazutsuna Yamaji, Yohei Hayashi

National Institute of Informatics

Kosuke Tanabe

National Institute for Materials Science

Toshihiro Aoyama

National Institute of Technology, Suzuka College

Takao Namiki, Hokkaido University

Daisuke Ikeda

Kyushu University

Outline

- Motivation
- Method
- Brief history of Repository Software
- 5 Trends of Repository software
- Discussion
- Conclusion

Motivation

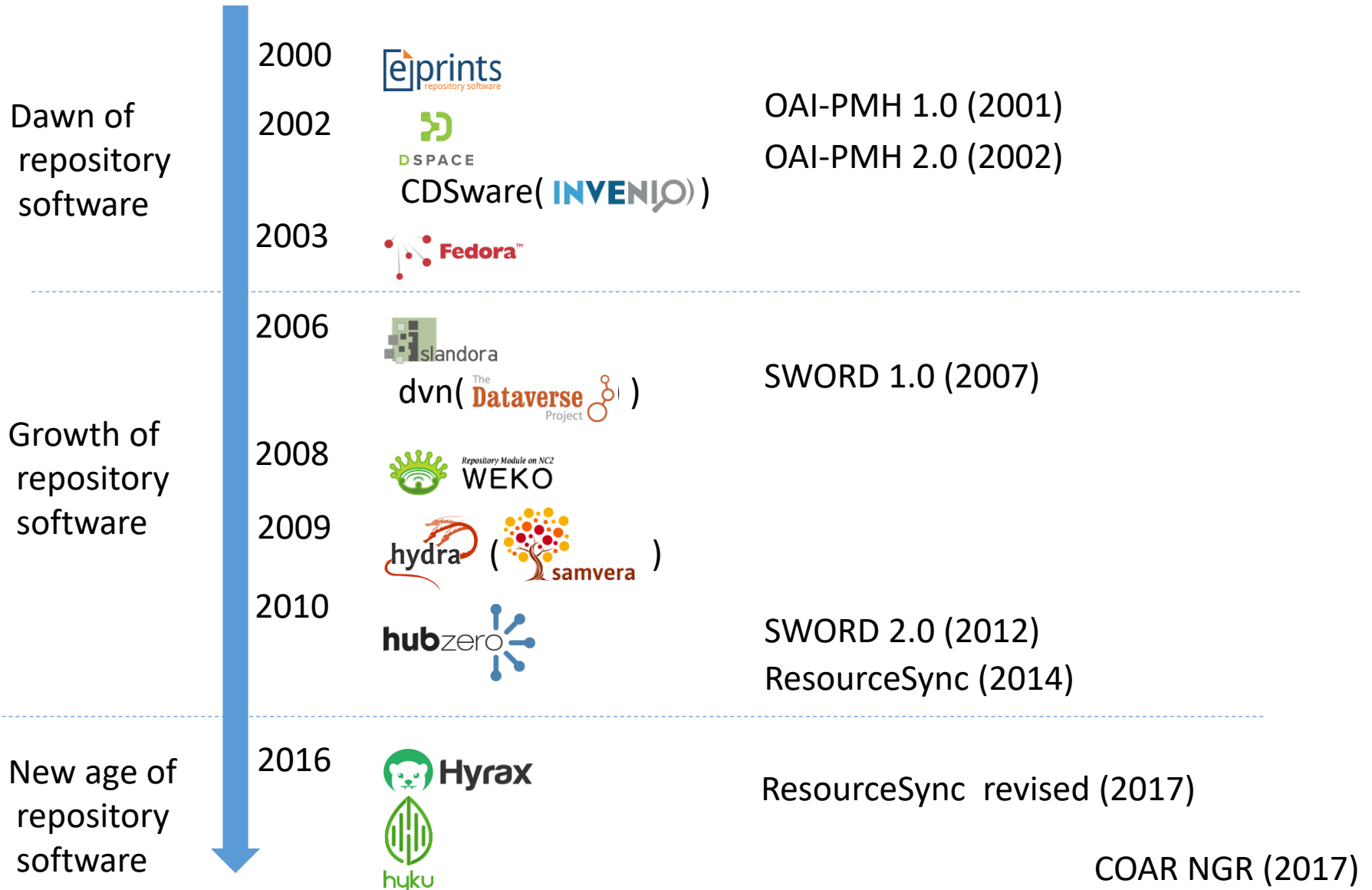
- In the near future, technical innovation of opensource repository software will be coming led by COAR Next Generation Repository.
- the repository manager and related people need to select repository software in consideration of technical trends and future prospects for building new repository.
- However, previous research mainly compared the functions of a specific version of repository without comparing the software release history and functions transition.
- In this research we compare the functions and release history of 9 types repository and aim to clarify technical trends and future prospects of the opensource-repository software.

Method

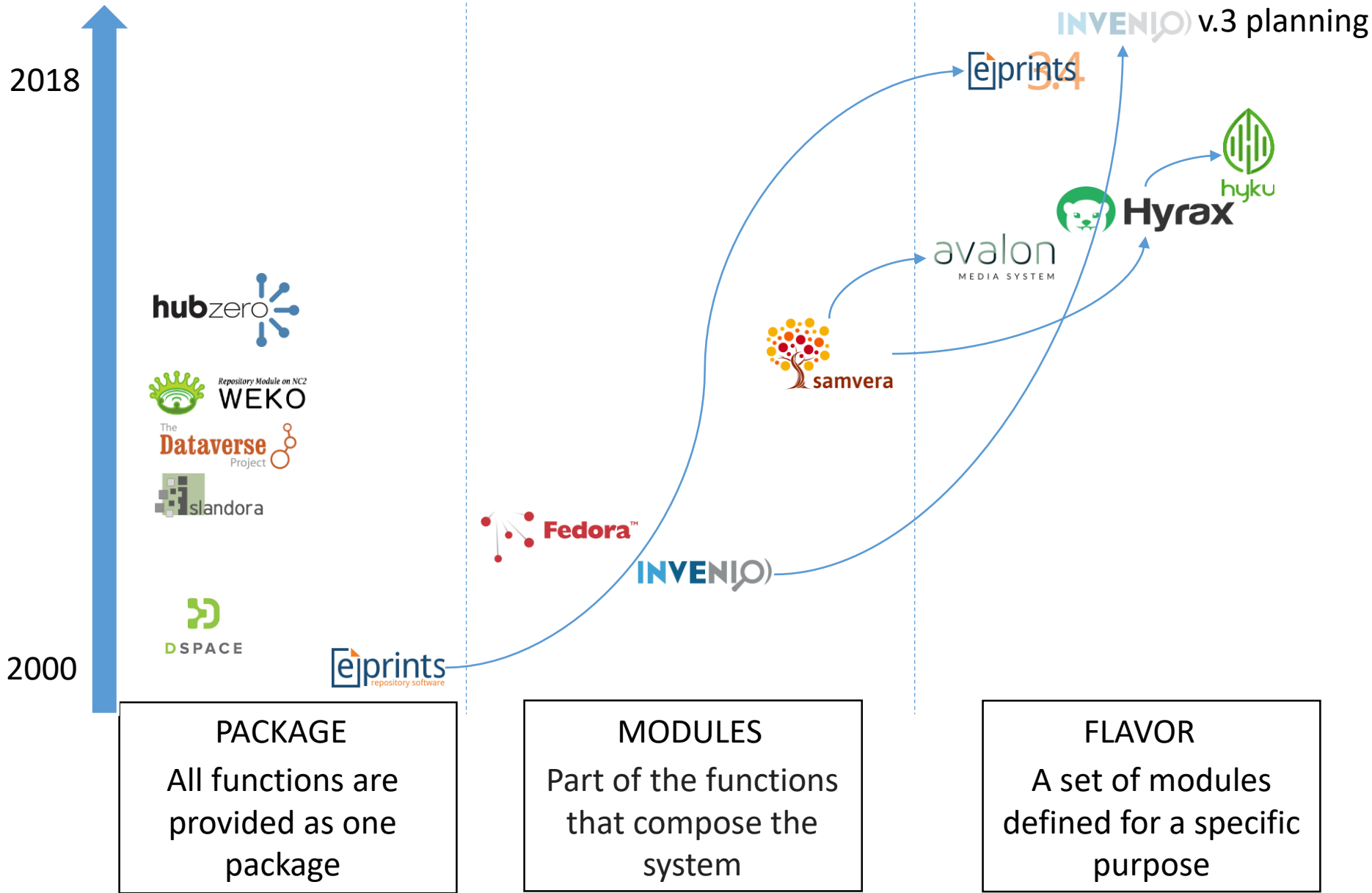
1. We chose 9 types of open-source repository software
 - DSpace
 - EPrints
 - Fedora
 - Islandora
 - HUBzero
 - Dataverse
 - WEKO
 - Samvera(Hyrax / Hyku)
 - Invenio(Zenodo)
2. We create history records of each open-source repository software
 - 3,800 history records are extracted from Paper, Wiki and Release Note
3. We create a comparison matrix of each open-source repository software technologies
 - Source: Wiki, source-code, running environment
4. We evaluate the history records and the comparison matrix

Result

Brief history of Repository Software



Trend: package, modules, flavors



Trend: Content Management System (CMS) based Repository

CMS based Repository

Islandora

v.1.0 (2009/4/3)

Drupal 5.x - 8.x

HUBzero

v.0.8.0(2010/9/15)

Joomla!

WEKO

v.1.0 (2008/3/28)

NetCommons2

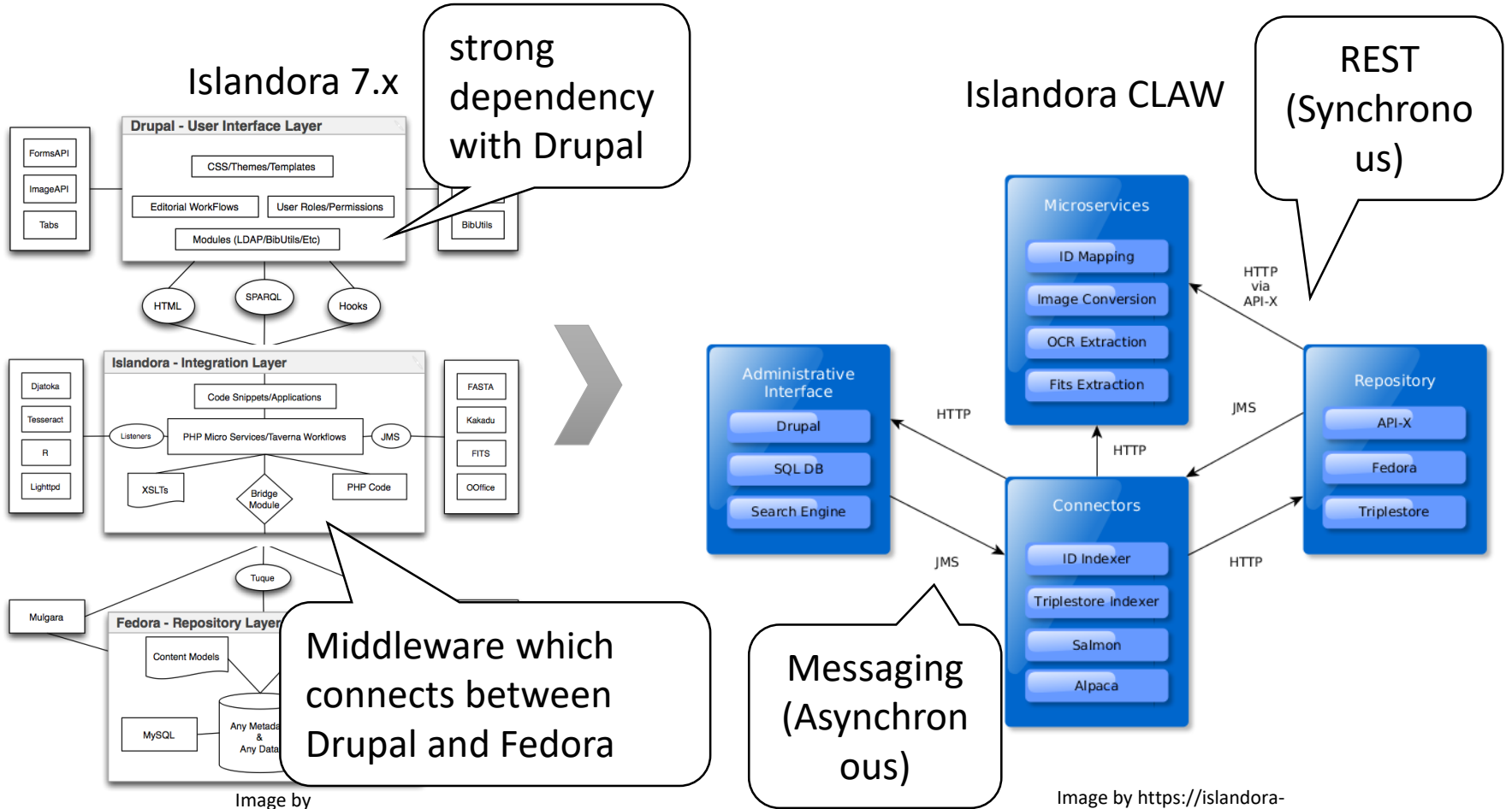
Pros and cons of CMS based Repository

Pros.	Cons.
Make it to manage web site and users easily	It's functions are restricted by CMS
Make it to extend functionality using modules in CMS	it has a strong dependency with CMS
Make it to support scholarly and social communication using modules in CMS	

Islandora CLAW gets independence and horizontal scalability by organizing the dependency (microservice architecture)

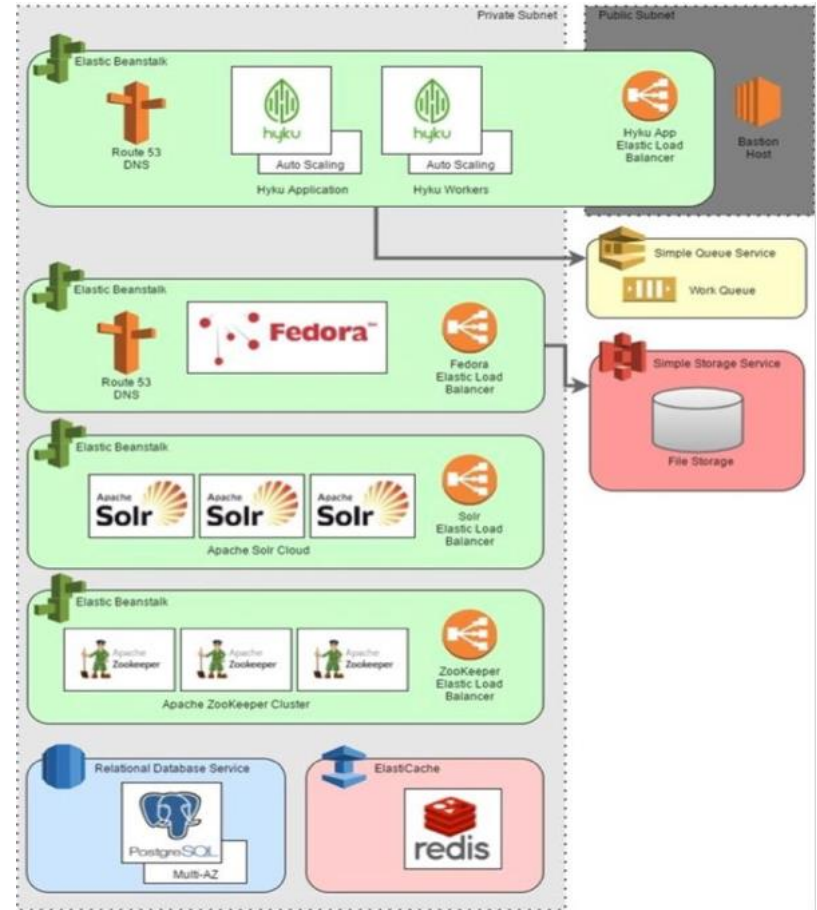
Islandora CLAW adopts microservice architecture

Microservice architecture is an architectural style that structures an application as a collection of loosely coupled services.



Trend: Standalone to Cloud

- Japan national repository service(WEKO)
 - Start : 2012/4
 - Organization: JPCOAR & NII
- DSpaceDirect (DSpace)
 - Start: 2014/03
 - Organization: DuraSpace
- HykuDirect (Hyku)
 - Start: 2017/2
 - Organization: DuraSpace
- No Hassle Hub (HUBzero)
 - Start: Unknown(could not find)
 - Organization: Purdue University
- EPrints Services (EPrints)
 - Start: Unknown(could not find)
 - Organization: University of Southampton



Hyku system architecture

Trend: Software Development

Name	Version	License	Version control	Issue tracker	CI	Mainling list	Chat	Vagrant	Docker	cloud deploy
DSpace6	6.2	BSD 3-Clause	GitHub	JIRA	Travis CI	Yes	IRC, Slack	Yes	-	-
DSpace7	-	BSD 3-Clause	GitHub	JIRA	Travis CI	Yes	IRC, Slack	-	-	-
EPrints3.3	3.3.16	GPL v.3	GitHub	GitHub	-	Yes	Gitter	Yes	-	-
Eprints3.4	3.4.0	GPL v.3	GitHub	GitHub	-	Yes	Gitter	-	-	-
Fedora 4	4.7.5	Apache 2.0	GitHub	JIRA, GitHub	Travis CI	Yes	IRC, Slack	Yes	-	-
HUBzero	2.1.15	GPL v.2	GitHub	Hubzero	Travis CI	-	-	Yes	-	-
Invenio3	3.0.0rc1	MIT	GitHub	GitHub	Travis CI	-	Gitter	Yes	Yes	OpenShift template
Zenodo	-	GPL v.2	GitHub	GitHub	Travis CI	-	Gitter	-	Yes	-
Hyrax	2.1.0.rc1	Apache 2.0	GitHub	GitHub	Travis CI	Yes	Slack	Yes	-	-
Hyku	v1.0.0.beta2	Apache 2.0	GitHub	GitHub	Travis CI	Yes	Slack	Yes	Yes	AWS template
Samvera	-	Apache 2.0	GitHub	GitHub	Travis CI	Yes	Slack	Yes	-	-
Islandra	7.x-1.10	GPL v.3	GitHub	JIRA	Travis CI	Yes	IRC	Yes	-	-
Islandora CLAW	-	GPL v.2	GitHub	GitHub	Travis CI	Yes	IRC	Yes	-	-
Dataverse4	4.8.6	Apache 2.0	GitHub	GitHub	Travis CI	Yes	IRC	Yes	Yes	OpenShift template
WEKO2	2.4.2	BSD 2-Clause	Bitbuket	-	-	Yes	-	Yes	Yes	-

Use of GitHub, Travis CI and Vagrant are de facto standards

Use of Slack, Gitter are spreading

Invenio, Hyku and Dataverse are providing cloud deploy template using AWS or OpenShift

Trend: System architecture

Name	Lang	Framework	Base	DataBase	O/Rmapping	Search	Full Text	REST API	Messaging	Hosting
DSpace6	Java	Spring	-	PostgreSQL, Oracle	Hibernate	Solr	Yes	Yes	-	Yes
DSpace7	Java	Spring	-	PostgreSQL, Oracle	Hibernate	Solr		Yes	-	-
EPrints3.3	Perl	-	-	MySQL		Internal	Yes	Yes	-	Yes
Eprints3.4	Perl	-	-	MySQL		Internal	Yes	Yes	-	Yes
Fedora 4	Java	Spring	ModeShape	ModeShape		Internal	-	Yes	-	-
HUBzero	PHP	Joomla!	-	MySQL		Internal	-	-	-	Yes
Zenodo	Python	Flask	Invenio3	SQLite, PostgreSQL, MySQL	SQLAlchemy	Elasticsearch	-	Yes	Celery/Rabbitmq	-
Invenio3	Python	Flask	-	SQLite, PostgreSQL, MySQL	SQLAlchemy	Elasticsearch	-	Yes	Celery/Rabbitmq	-
Hyrax	Ruby	Rails	Samvera	SQLite, PostgreSQL, MySQL	ActiveFedora	Solr	Yes	Yes	Sidekiq/Redis	-
Hyku	Ruby	Rails	Hyrax	SQLite, PostgreSQL, MySQL	ActiveFedora	Solr	Yes	Yes	Sidekiq/Redis	Yes
Samvera	Ruby	Rails	Fedora4	-	ActiveRecord	Solr	-	Yes	-	-
Islandra	PHP	Drupal7	Fedora3	MySQL	Doctrine	Solr	Yes	Yes	-	-
Islandora CLAW	PHP	Drupal8	Fedora4	MySQL	Doctrine	Solr	-	Yes	API-X(Karaf), ActiveMQ	-
Dataverse4	Java	primefaces	-	PostgreSQL	Hibernate	Solr	Yes	Yes	-	-
WEKO2	PHP	NetComm ons2	-	MariaDB		Mroonga	Yes	-	-	Yes

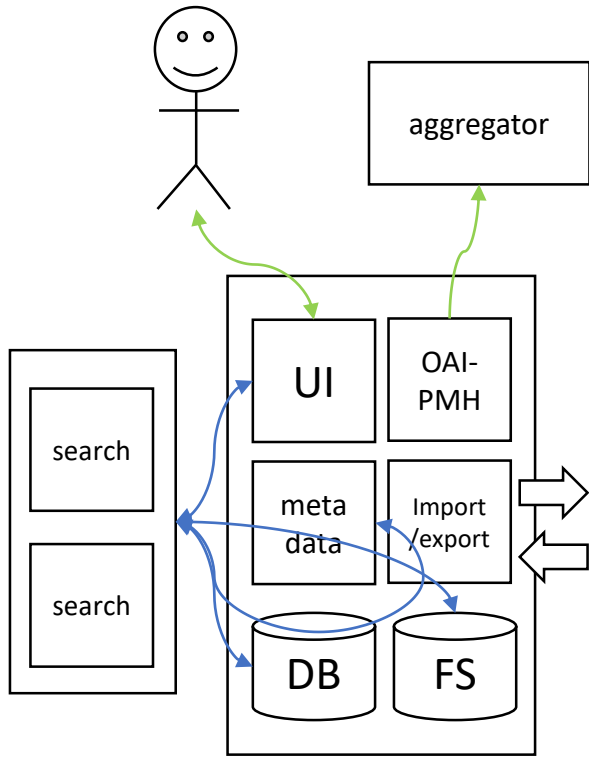
Software development has become complicated

However, automating build/test/deployment with CI/CD tool can help developers

Container orchestration software will improve maintenance and operability

Toward Next Generation Repository

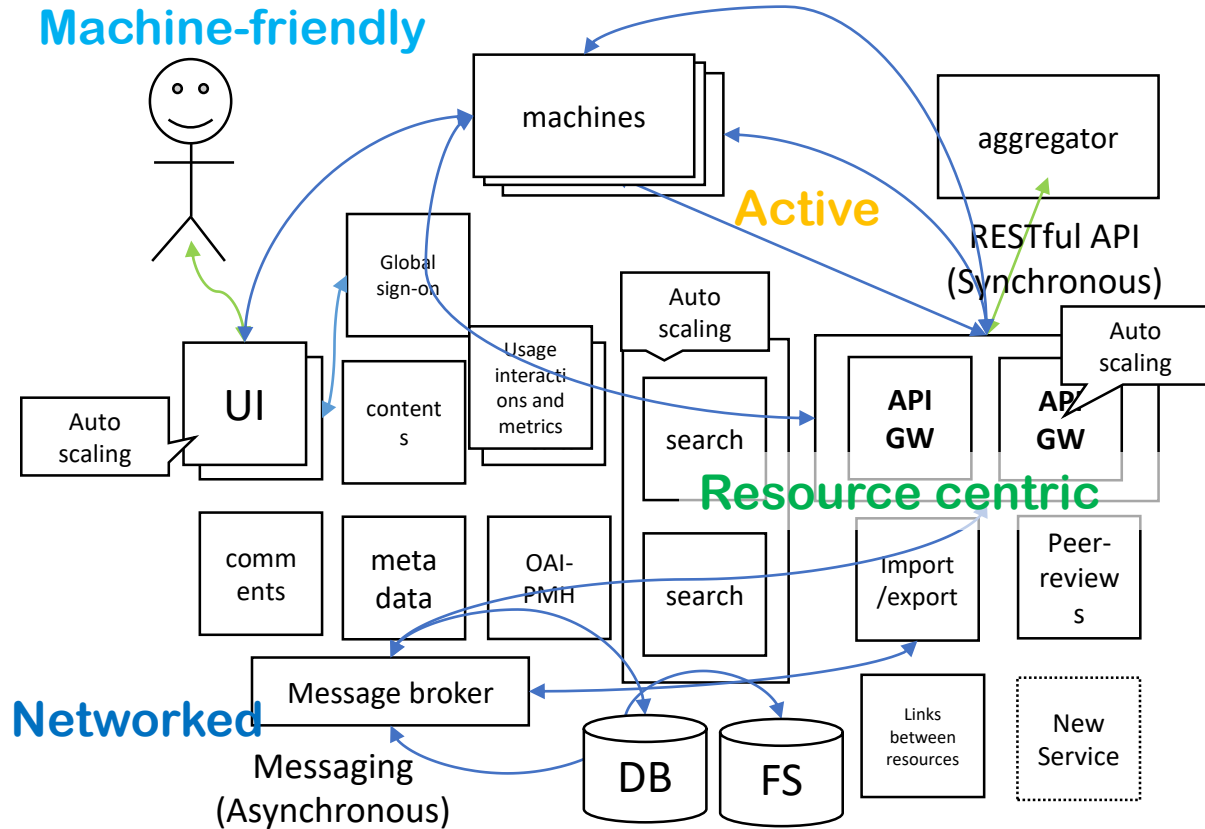
Current repositories



based on

a monolithic architecture
(provides all functions in one system)

Next generation repositories



will be based on

cloud/microservice architecture
API gateway pattern will be required
(like a Fedora's API-X)

Conclusion

- In this presentation, we evaluated history and system architecture of repository software, obtained the following technology trends.
 - Trend: package, modules, flavors
 - Trend: Content Management System (CMS) based Repository
 - Trend: Standalone to Cloud
 - Trend: Software Development
 - Trend: System architecture

Microservice architecture
(especially API gateway pattern*)
becomes an important technology in a future

Thank you

Masaharu Hayashi
mhaya@nii.ac.jp