



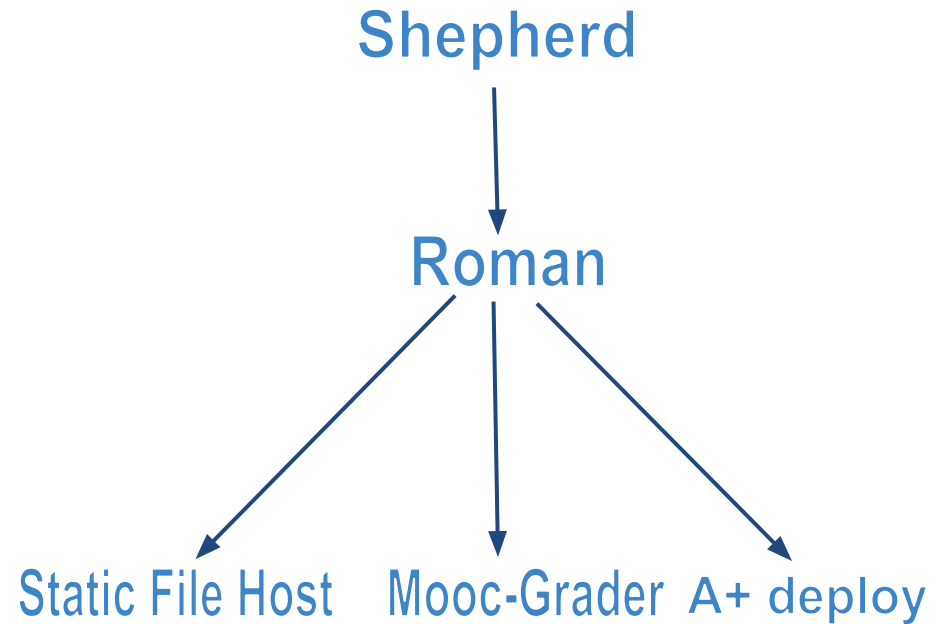
Aalto University

A+ Shepherd

Qianqian Qin
29.04.2020

Tools for course management and course building

- **Shepherd**
 - *Course management service*
 - *Trigger the automation process*
- **Roman**
 - *Build the course*
 - *Run docker containers*
- **Deployment**
 - *Docker container for uploading files*
 - *Docker container for aplus deploy*
 - *Static File Host Server*
 - *Mooc-grader deploy APIs*
 - *A package for transferring files*



Shepherd

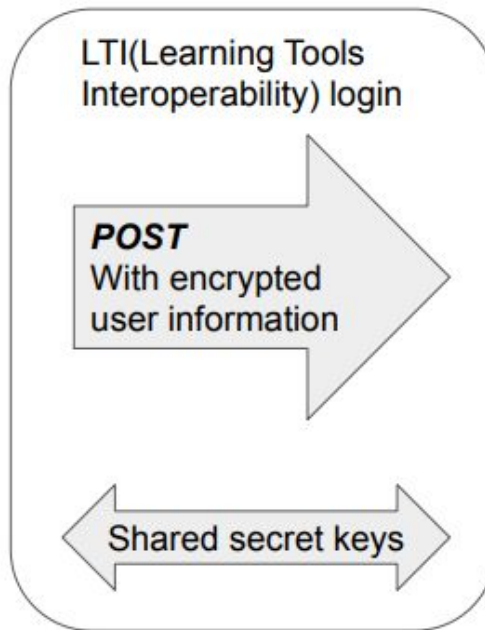
- Oauth: LTI Login
- Group and permission management
- Course management
- Automation of course build and deployment

Shepherd - LTI Login



Course staff

- Participants
- Groups
- All results
- Visualizations
- Edit news
- Edit course
- Shepherd**



Shepherd

You are leaving A+ and entering Shepherd

- Information including your name and email address will be sent to the service. This means that at least the service's administrators can identify you. You can see the full list of information below. We use this external service to help you and other students learn. Nevertheless, you aren't absolutely required to use it in order to pass the course. If you want to avoid using the service, please discuss alternative solutions with the course staff.
- Moreover, we send this service a so-called **access token**, which gives the service access to the A+ API at your privilege level. If you don't want that to happen, please contact A+ administration.
- This service is not covered by any privacy regulations, thus there can't any lawful entity to enforce it. Please see the **service's privacy notice** for more details.

You can first take a look at the external service without sending data: [service's front page](#).

Don't show this and automatically continue to the service next time.
This choice is saved on your device and you can remove it on your profile page.

[Continue to the service](#) [Show details](#)

```
POST http://127.0.0.1:5000/
context_id 127.0.0.1:8000/default
context_label DEF000
context_title Def. Course
custom_context_api http://127.0.0.1:8000/api2/courses/1/
```

Shepherd

- Group and Permission Management

Group

- Structure: Tree-like model
- Implement *sqlalchemy_mptt* package
 - Hierarchical model
 - implementing Modified Preorder Tree Traversal
 - efficient retrieval (insert and remove involved more)
- User - Group: Many to Many

Shepherd

- Group and Permission Management



Shepherd

[Home](#)

[Courses](#)

[New Course](#)

[Groups](#)

[My groups](#)

[My Repos](#)

Group list

[Create group](#)

- *aalto* [Members](#) [Subgroups](#)
 - *sci* [Members](#) [Subgroups](#)
 - *cs* [Members](#) [Subgroups](#)
 - *csb* [Members](#) [Subgroups](#)
 - *programming_1* [Members](#) [Subgroups](#)

Shepherd

- Group and Permission Management

Group Permission

- Group permission
 - Create subgroups
 - Create courses
- Create group
 - Create subgroups under a allowed target group
- Create course
 - Create courses with a specific prefix (CS-)
- Manage course
 - A user with the administrator permission with the course can manage the course
 - Assistant role could not

Shepherd - Course Management

User id:1

Course Key	Instance Key	Name	Identity	Git	Operations				
CS-A1000	2020	Programming	cs	https://version.aalto.fi/gitlab/qinq1/course-templates.git:master	Edit	Duplicate	Owner groups	Obtain JWT token	Delete
CS-A1001	2020	Web Development	cs	https://github.com/QianqianQ/course-templates.git:master	Edit	Duplicate	Owner groups	Obtain JWT token	Delete

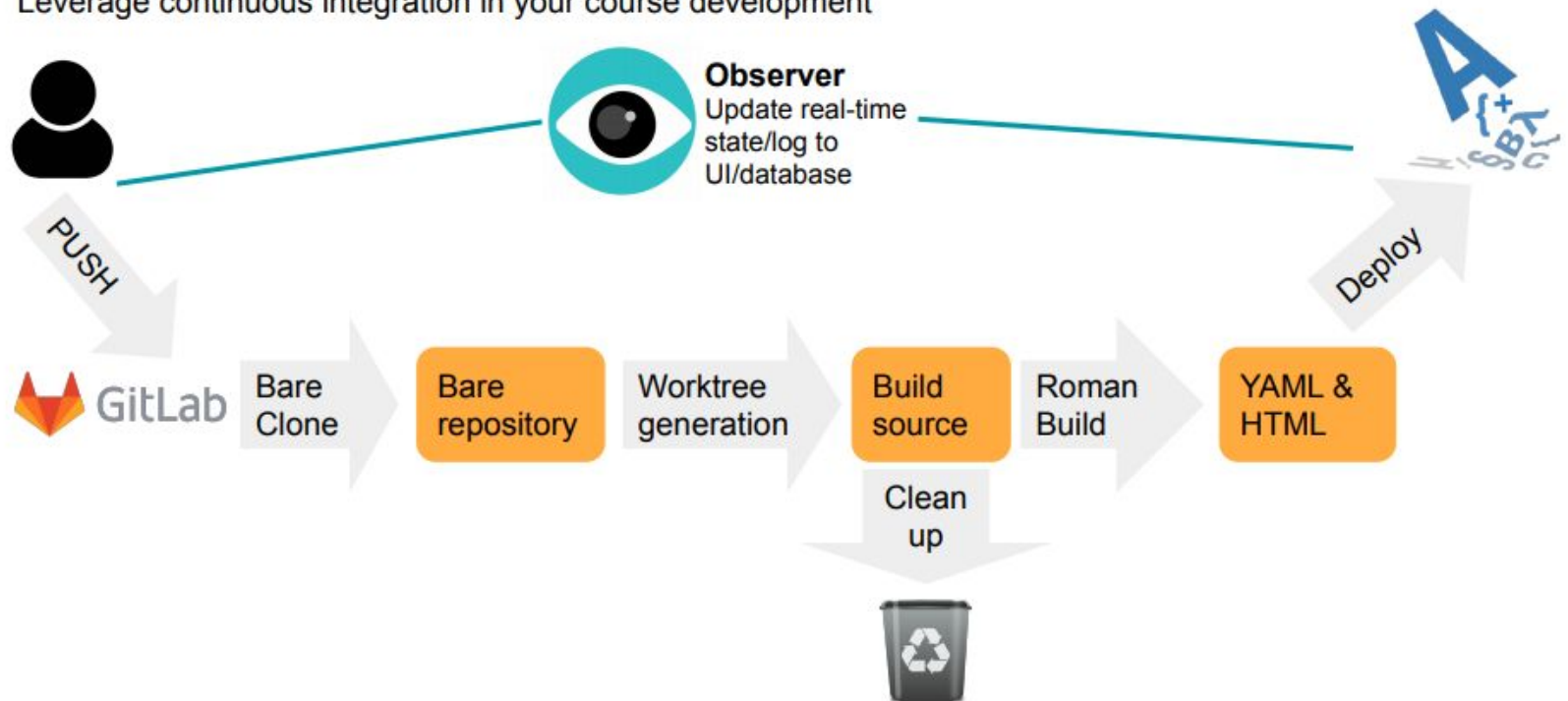
Courses

Shepherd

- Automation of course build and deployment

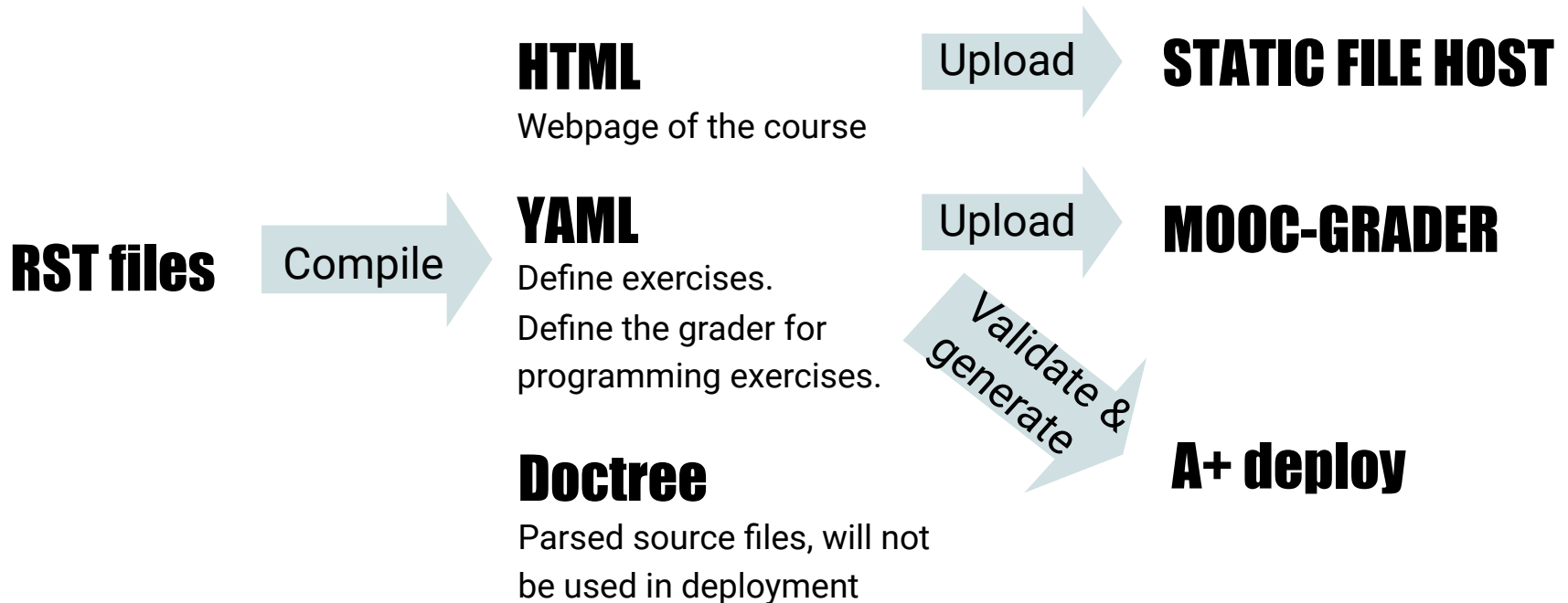
All you need is “Push”

Leverage continuous integration in your course development



Roman

- Compiling course materials from RST files
- Reads and runs steps from a config file (course.yml)
- Validate the config files (yamllidator package)

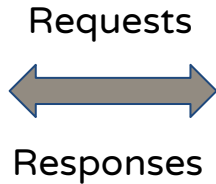


Deployment

- **Static File Host Server**
 - *Host the static files*
- **Mooc-grader deploy APIs**
 - *Deploy compiled yaml files to mooc-grader*
 - *Update index.yaml for A+ deploy*
- **Docker container for deploying files**
 - *Run scripts for deploying files to servers*
 - *yaml files to mooc-grader*
 - *static files to static file host*
- **Docker container for aplus deploy**
 - *Validate yaml files and generate A+ json for future usages*
- **Package for transferring files**
 - *Helpers for file deployment*

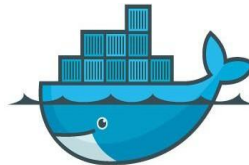
Static File Host - Structure

Client



Reverse Proxy

NGINX



docker

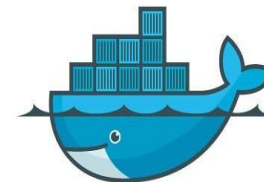
Server



uWSGI



Flask



docker

Docker-compose
Shared volume
Retrieve static files from nginx

Static File Host & Mooc-grader (deploy) APIs

Endpoints:

- **/`<course_name>`/select-files**
 - determine whether and which files to upload
- **/`<course_name>`/upload-files**
 - upload the files to the server (a temp directory with a unique id)
- **/`<course_name>`/publish-files**
 - publish the newly uploaded files (to the course directory)

Extras in Mooc-grader:

- **/`<course_name>`/delete**
 - Delete the files of a course
- **/`<course_name>`/files/`<relative_file_path>`**
 - Delete a specific file of a course
- **/`<course_name>`/update-index-file**
 - Update index.yaml for A+ deploy

Deployment - Authentication

Authorization in Headers: **JSON Web Token (JWT)**

- JWT of a course generated in Shepherd
- Servers shared a public key
 - Shepherd, Static File Host, Mooc-grader
- Decode JWT, if success:
 - check the payload: `course_name` matches the course name in the url

Example of JWT public key

```
JWT_PUBLIC_KEY = ""
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCQAQ8AMIIBCgKCAQEAA0QIB6wP5rGpT7pckM0uQ
bn3FbQI2Xp58vLW+eLISgPvh0EMNuVVMazRFTBGnSxYI2P2F+Yf+08ck3JW0puCD
+i0a+RlC7gZdspULHpRYScc0qvRdcMn93nuPxiHJ+zAFuVR6mmDQmkHR3ruFvbQt
FWABpbZpqV0la0UqoyQcp7JG0rrgZZhifS8EE56azvhIm8n2qf+KhKkTq0P71j+4
3h2sZtHM9nrsm/wtyb26xPBwGS1v1d5bWw0D2vhPSCP4HV2DuI6Wd6pEN9Axjf5j
dG7tGa6GnyPchdDAvlnA1FQiFfkz4NQtL5upmGiz6gBs1F1PhZmej1r2RUYd4mbQ
3QIDAQAB
-----END PUBLIC KEY-----
""
```

Example of payload

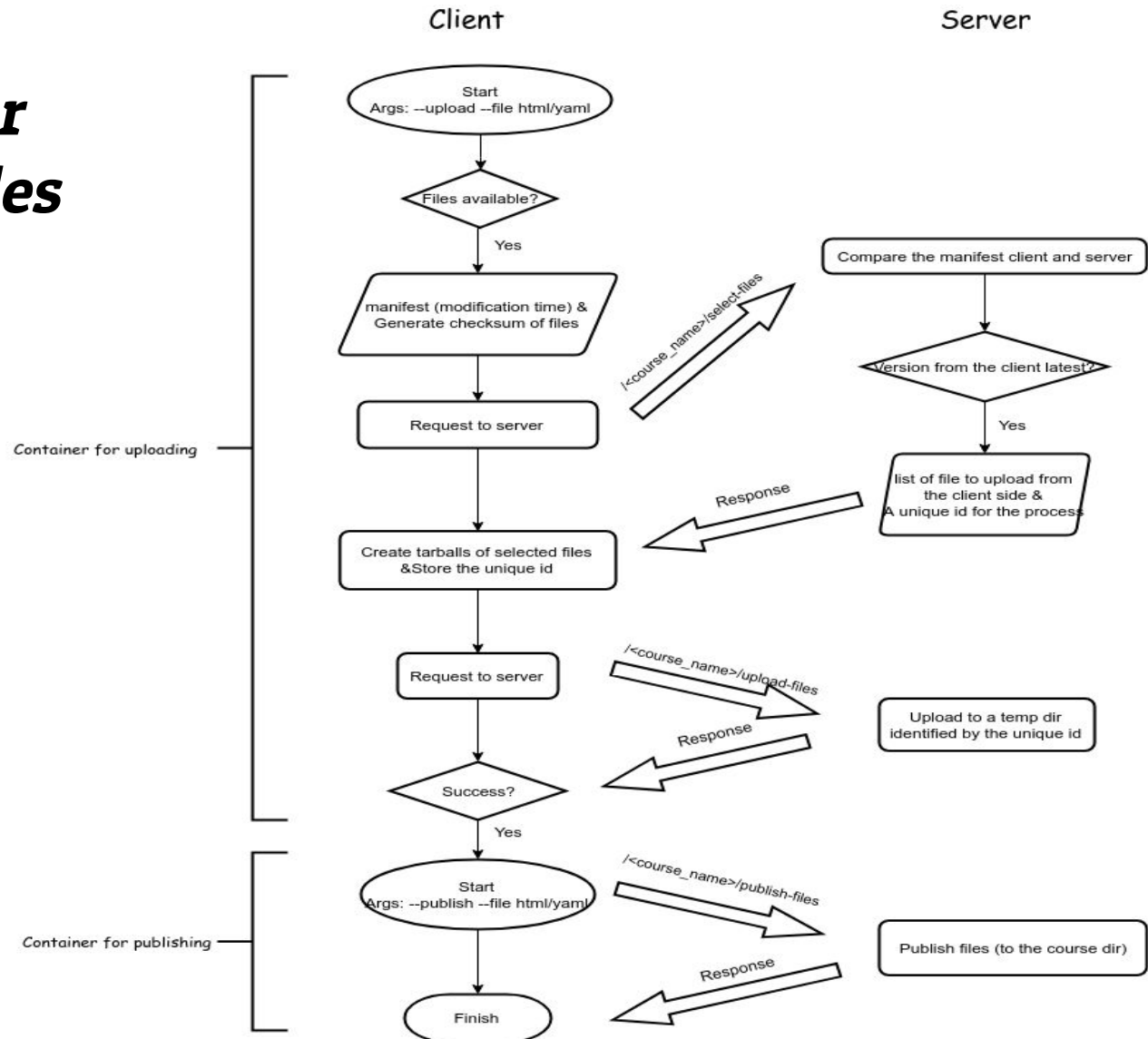
```
{
  'sub': 'def_course',
  'iss': 'shepherd',
  'iat': 1562828304
}
```

Docker container for deploying files

- **Environment Variables:**
 - PLUGIN_API -- API Base url
 - PLUGIN_COURSE -- course name
 - PLUGIN_TOKEN -- JWT Token
- **Command Line Arguments:**
 - exclusive actions:
 - --upload / --publish
 - file type: --file -f
 - yaml / html
- **Run two containers for one deployment:**
 - First one with --upload
 - Second one with --publish

```
#!/usr/bin/env bash
docker run --rm -it --name upload_files --network="host" \
-w /data/ \
-v "$(pwd):/data/" \
-e PLUGIN_API=http://0.0.0.0:5001/ \
-e PLUGIN_TOKEN=eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdWIiOiJKZWZfY291cnNlIiwiaWF0IjoxNTYyODI4MzA0L... \
-e PLUGIN_COURSE=def_course \
apulsms-file-transfer-client --upload -f html && \
echo "upload static files successfully" && \
docker run --rm -it --name publish_files --network="host" \
-w /data/ \
-v "$(pwd):/data/" \
-e PLUGIN_API=http://0.0.0.0:5001/ \
-e PLUGIN_TOKEN=eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdWIiOiJKZWZfY291cnNlIiwiaWF0IjoxNTYyODI4MzA0L... \
-e PLUGIN_COURSE=def_course \
apulsms-file-transfer-client --publish -f html && \
echo "publish static files successfully"
```

Workflow for deploying files

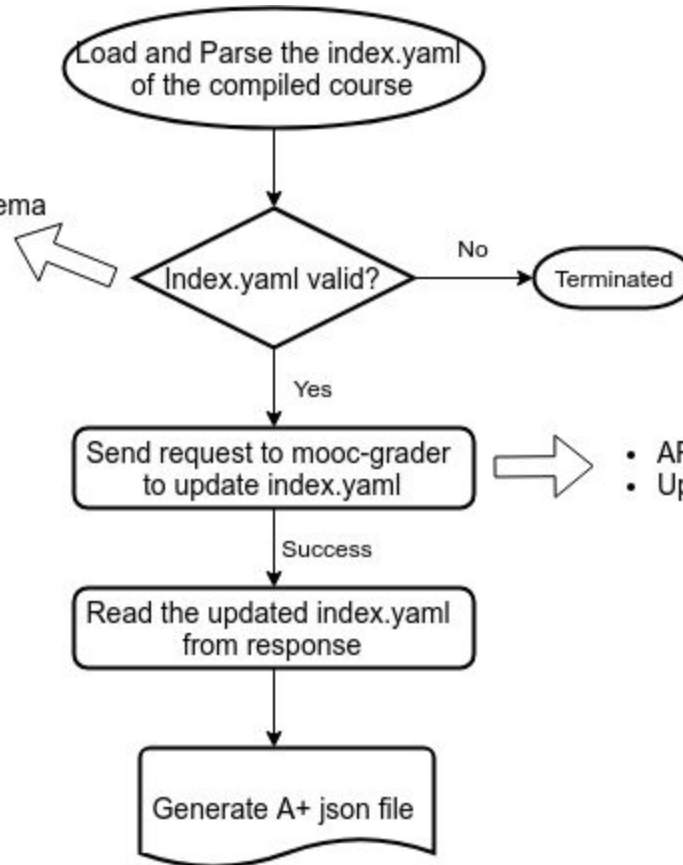


Package for transferring files

- Functions for both server side and client side
 - For containers and the servers
- Multiple uploading methods based on the file size
 - ≥ 50 MB: compressed one by one
 - < 50 MB: collected to fill a quota (50 MB) and then compressed
 - compressed file:
 - ≤ 4 MB post directly -- form-data
 - > 4 MB: post by chunks -- octet-stream
- Specific authentication and upload functions for Django and Flask
 - Different interfaces accessing requests
 - Different interfaces accessing application setting
- Could be more generic

Deployment - Docker container for A+ deploy

- Yamlidator package
- Predefined customed json schema



- API: /<course_name>/update-index-file
- Update url fields of the exercise and static content

Deployment - Example

<https://apluslms.github.io/events/presentations/2020-04-29-roman-run-qin.mp4>

THE END

THANKS!