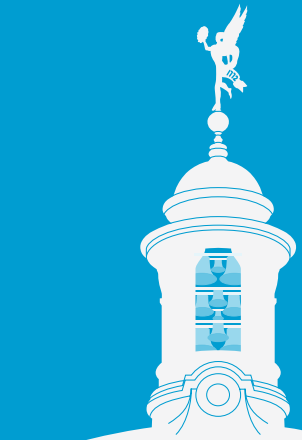
WWU
MÜNSTER

Integrating Moodle with other educational software

Jan Dageförde and Tobias Reischmann, University of Münster

October 1, 2019



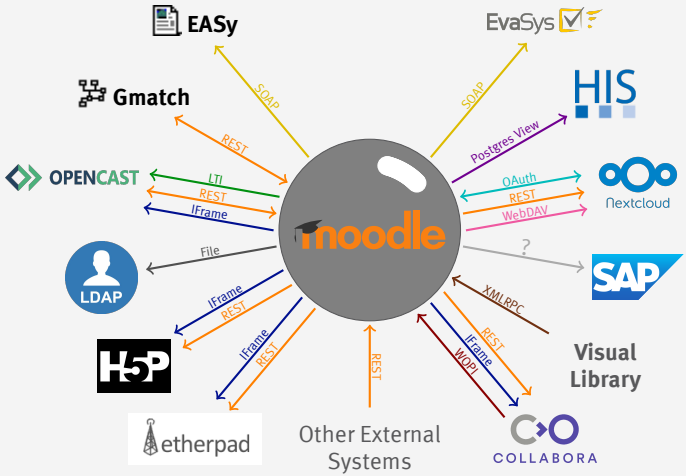
living.knowledge

ZHLdigital

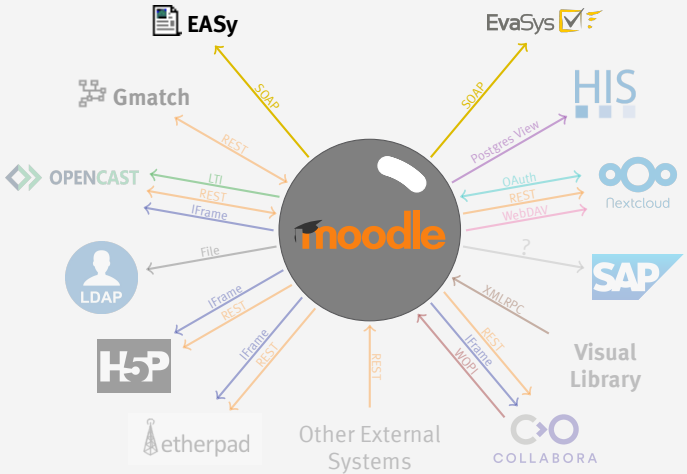
“Can we use Moodle …

- as a campus management system,
- for lecture evaluation,
- for tracking progress of master theses,
- to coordinate recognition of grades from a semester abroad,
- …?”

Integration technologies



Integration technologies



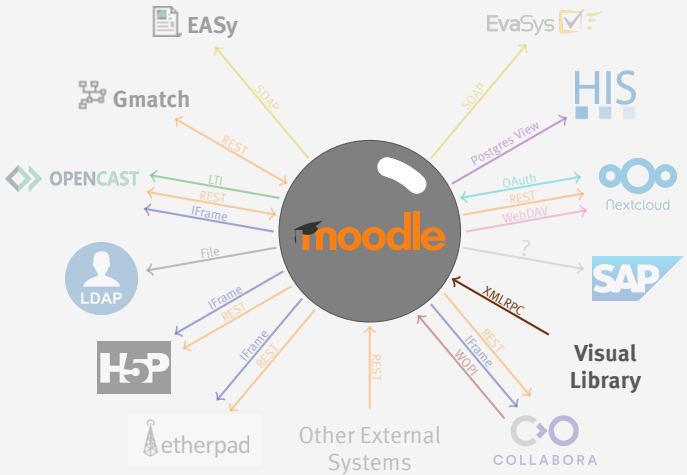
SOAP Webservice

- Offer and query web resources
- Definition and requests through XML-documents
- Authentication through user/password or token

- Incoming Docs: https://tinyurl.com/moodleWSdoc
- Outgoing Docs: https://tinyurl.com/soapoutc

Requires `soap` extension

Integration technologies



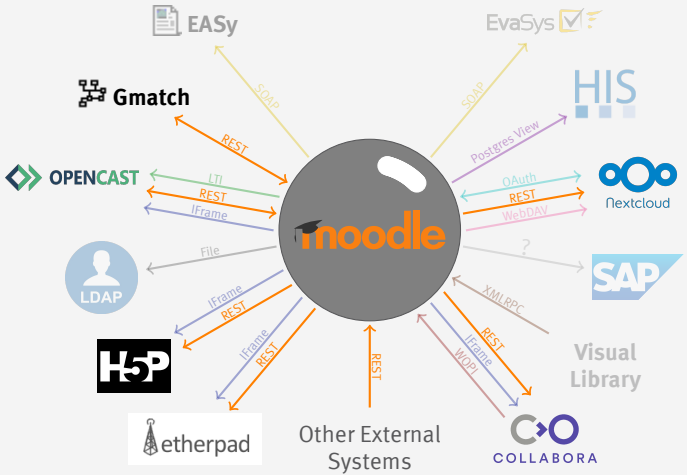
XML-RPC Webservice

- Perform a remote procedure call
- Call methods instead of transferring resources

- Incoming Docs: https://tinyurl.com/moodleWSdoc
- Outgoing Docs: https://tinyurl.com/xmlrpcout

Requires `xmlrpc` extension (experimental since 2001 / PHP 4.1.0!)

Integration technologies



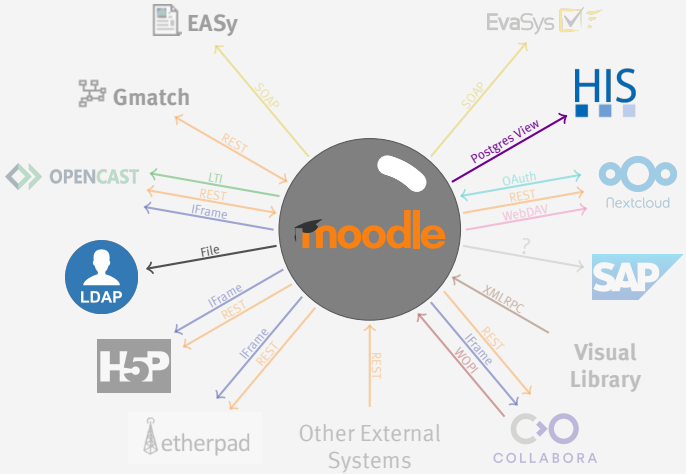
RESTful Webservice

- Offer and query web resources
- Stateless = No user session
- Authentication through user/password or token

- Incoming Docs: https://tinyurl.com/moodleWSdoc
- Outgoing Docs: https://tinyurl.com/restout

Requires `curl` extension

Integration technologies



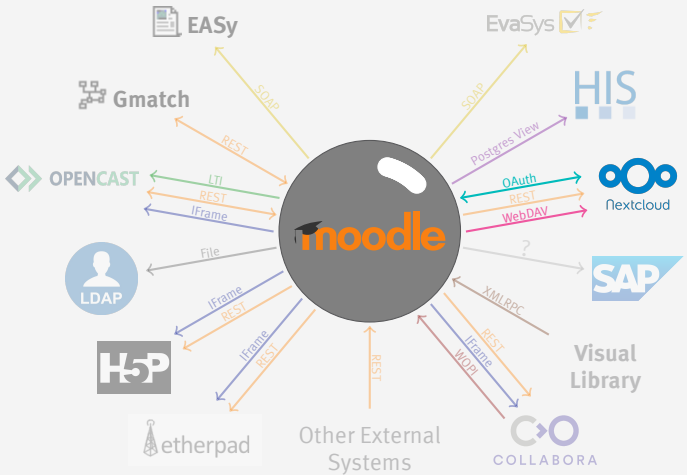
Postgres view

- Read-only database tables
- Makes second database connection necessary
- Postgres View: https://tinyurl.com/postgresview
- Access in PHP: https://tinyurl.com/pgsqlphp

File

- Both systems access the same file on local/remote storage
- One side should be read-only

Integration technologies



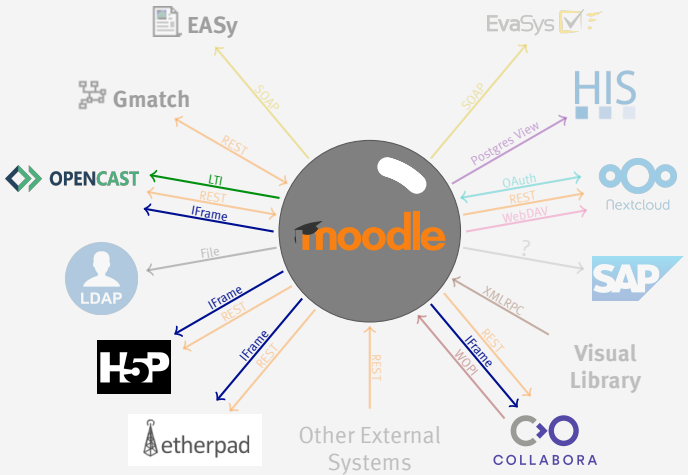
OAuth 2.0 (Related: OpenID)

- Request authorisation to access system
- Token as identifier
- Moodle OAuth 2 API: https://tinyurl.com/oauth2api

WebDAV

- File browsing and manipulation
- Moodle WebDAV client: https://tinyurl.com/webdavlib

Integration technologies



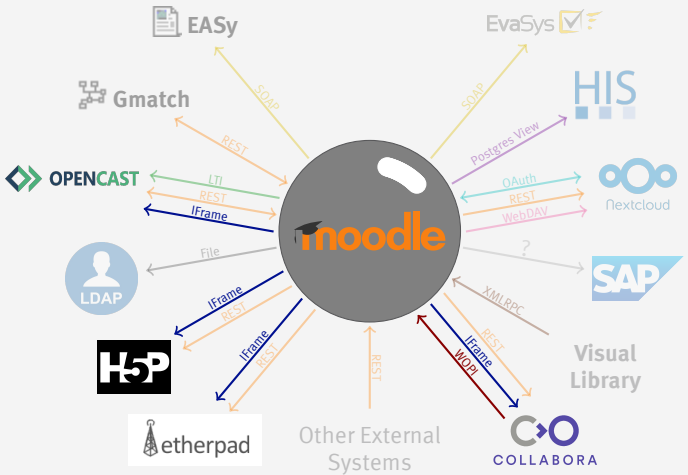
<iframe

- Embed external views
- Seamless authentication: LTI or tokens as parameters

LTI

- Learning Tools Interoperability
- e. g. as authentication against external system
- Moodle LTI Client: https://tinyurl.com/moodleLTI

Integration technologies



<iframe

- Embed external views
- Seamless authentication: LTI or tokens as parameters

WOPI

- (Single) file manipulation
- Office 365, Collabora Online as clients
- Custom WOPI Host implementation: https://tinyurl.com/customwopi

“I want to integrate a system – What do I do?”

- What are you trying to achieve?
- What integrations does the other system offer?
- Do you need read or write access?
- Which methods of authentication does the system offer?
- Might your integration cause particular high load for any system?

Benefits

- Best-of-breed
- Increase performance by using other implementation languages
- (Hopefully) seamless interaction with external systems

Potential issues

- Added latency
- Added complexity
- API changes

Lessons learned (1/3)

Performance/Latency

- Caching of responses
- Keep external systems from critical path
- Do not query external systems on every page load

Security considerations

- Do not store (personal) credentials
- Restrict systems' privileges for accessing Moodle
- Do not hit external systems too frequently

Lessons learned (2/3)

External APIs

- Stable API required
- Ideally, API is versioned
- Consider querying version of the external API

```
/ocs/v1.php/apps/files_sharing/api/v1/shares
Accept: application/v1.0.0+json
```

Software Testing

- Consider: Automated integration testing with Docker
- Otherwise, mock external system with API specs

Lessons learned (3/3)

Active communication is crucial!

- Administrators of the external system
- Developers of the original software

Publish to the plugins directory

- Sharing is caring
- Add detailed instructions for reviewers