“The **impact** we cherish is **discovery** in science”

Randy Schekman,
eLife Editor-in-chief

eLIFE

Introducing eLife Continuum

Webinar – August 4, 2016

4:00 – 5:00pm UK

Ian Mulvany

eLife Head of Technology

Giulano Maciocci

eLife Head of Product

Siân Roderick

eLife Web Product Manager

Agenda

- Introductions
- eLife journal on eLife Continuum platform
- eLife Continuum demo
- Next steps – use and further development
- Time for questions

Please type your questions in the questions panel.

Note that we will endeavour to answer those in the time provided in the latter part of the session.

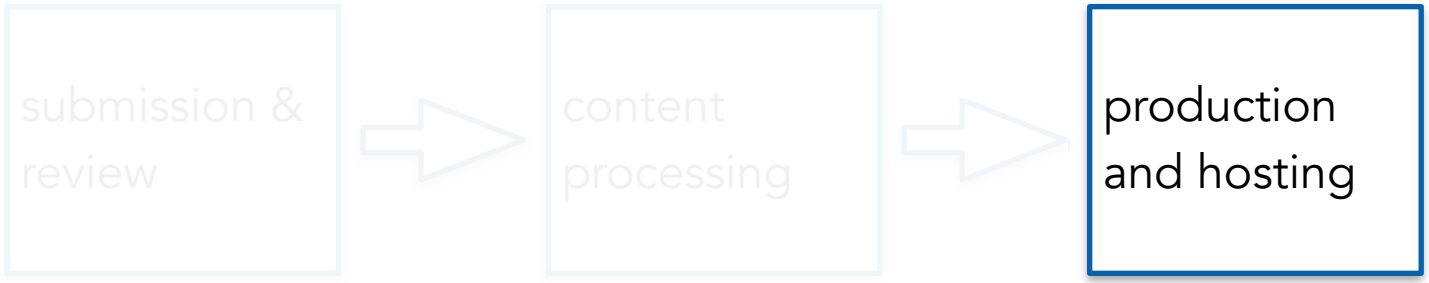

eLIFE
Helping scientists accelerate discovery by
operating a platform for research communication
that encourages and recognises the most
responsible behaviours in science

- What we did
- Motivation
- Key Features
- Limitations
- Architecture
- Amazon Web Services Dependencies
- Code Structure
- Installation and Deployment
- Demo
- Feature Requests and Feedback
- Time for questions

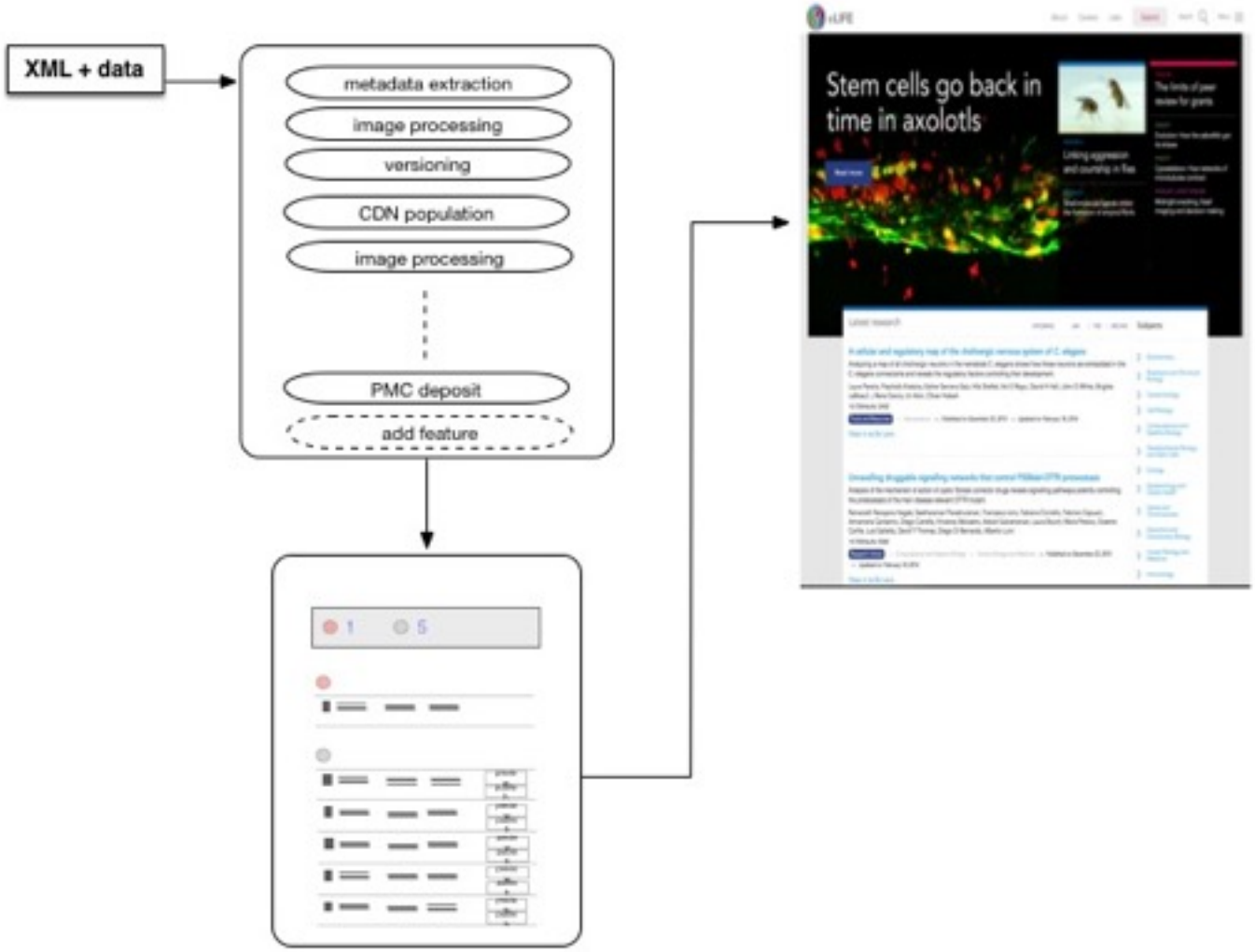
- **What we did**


submission & review
content processing
production and hosting

- **What we did**


submission & review
content processing
production and hosting

- **What we did**


XML + data
metadata extraction
image processing
versioning
CDN population
image processing
PMC deposit
add feature
Stem cells go back in time in axolotls

- **Motivation**

- Provide control over our content, end to end
- Support future developments on our platform
- Contributes towards the creation of an ecosystem of open scholarly infrastructure

We are happy because:

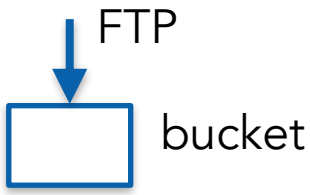
- Just not that many open source pieces of infrastructure for STM publishing in the wild
- It's actually possible to run and configure multiple versions
- It's almost trivial to extend the back-end functions
- The dashboard auto-extends to show any data you feed it

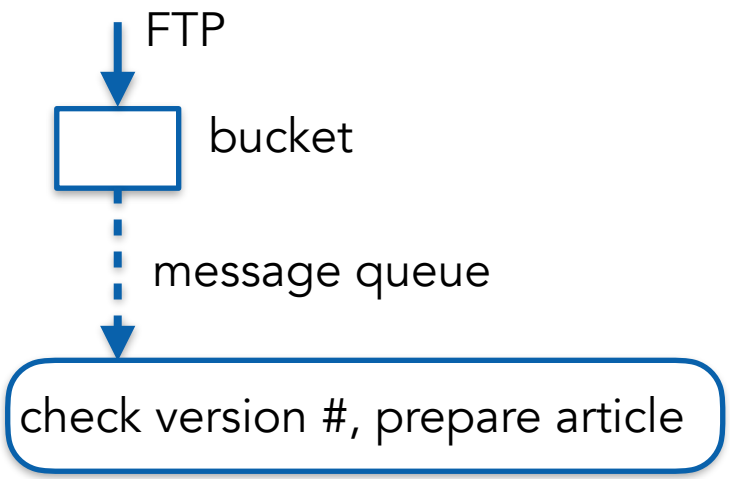
- **Key Features**
  - Good previews of articles ahead of publication
  - Article versions
  - Scheduling for future publication
  - Modular and extensible
  - Works with JATS XML

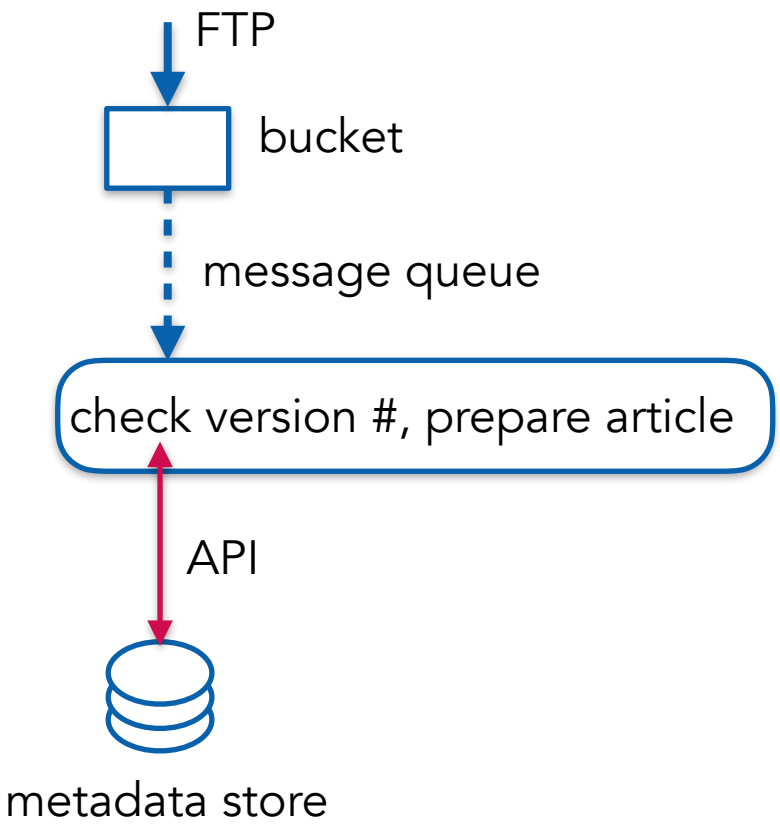
- **Limitations**

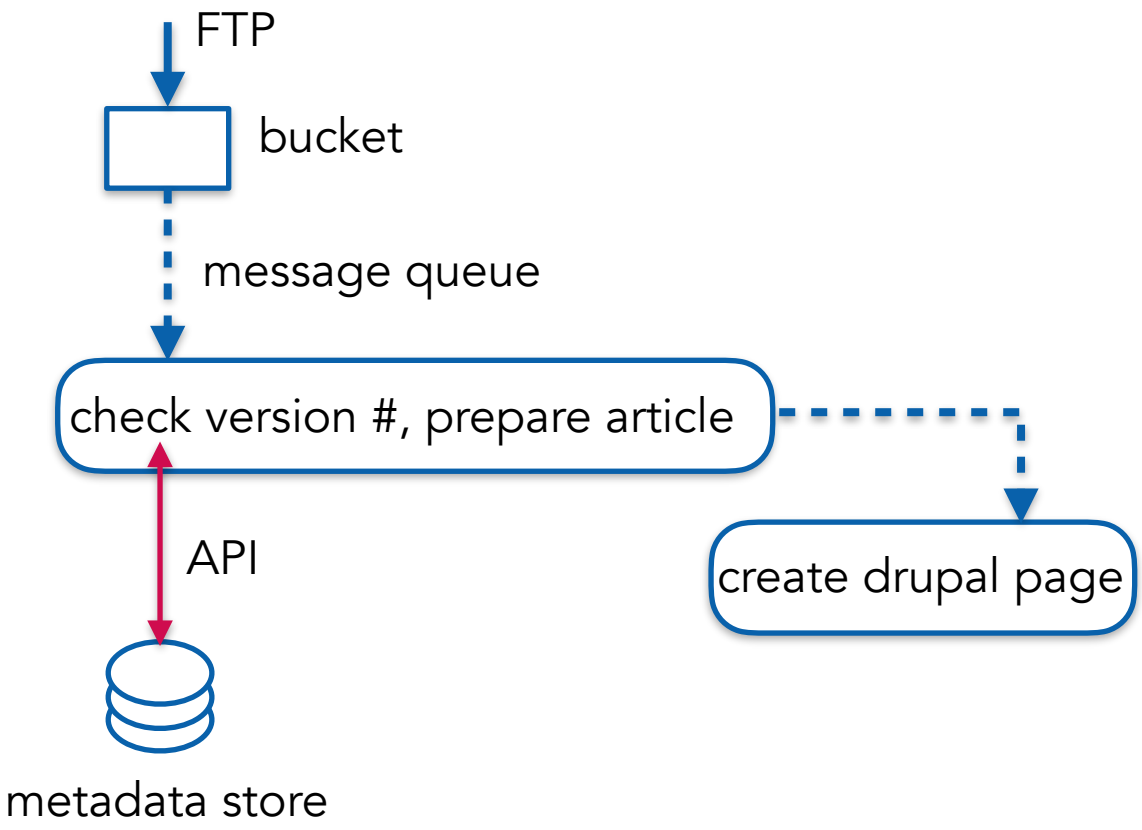
  - Does not support issues
  - Currently only explicit support for one journal
  - Production system does not support user accounts
  - Installation is quirky

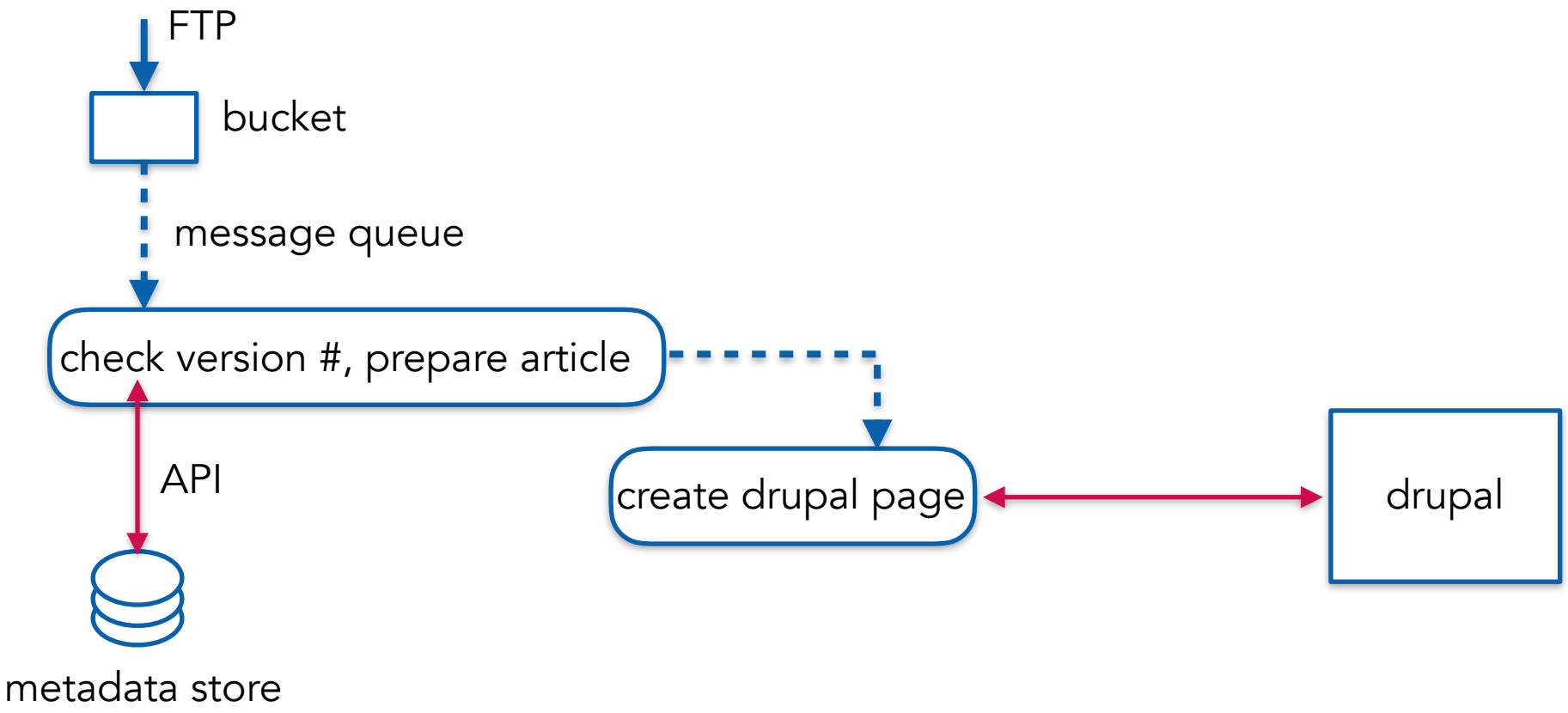
bucket

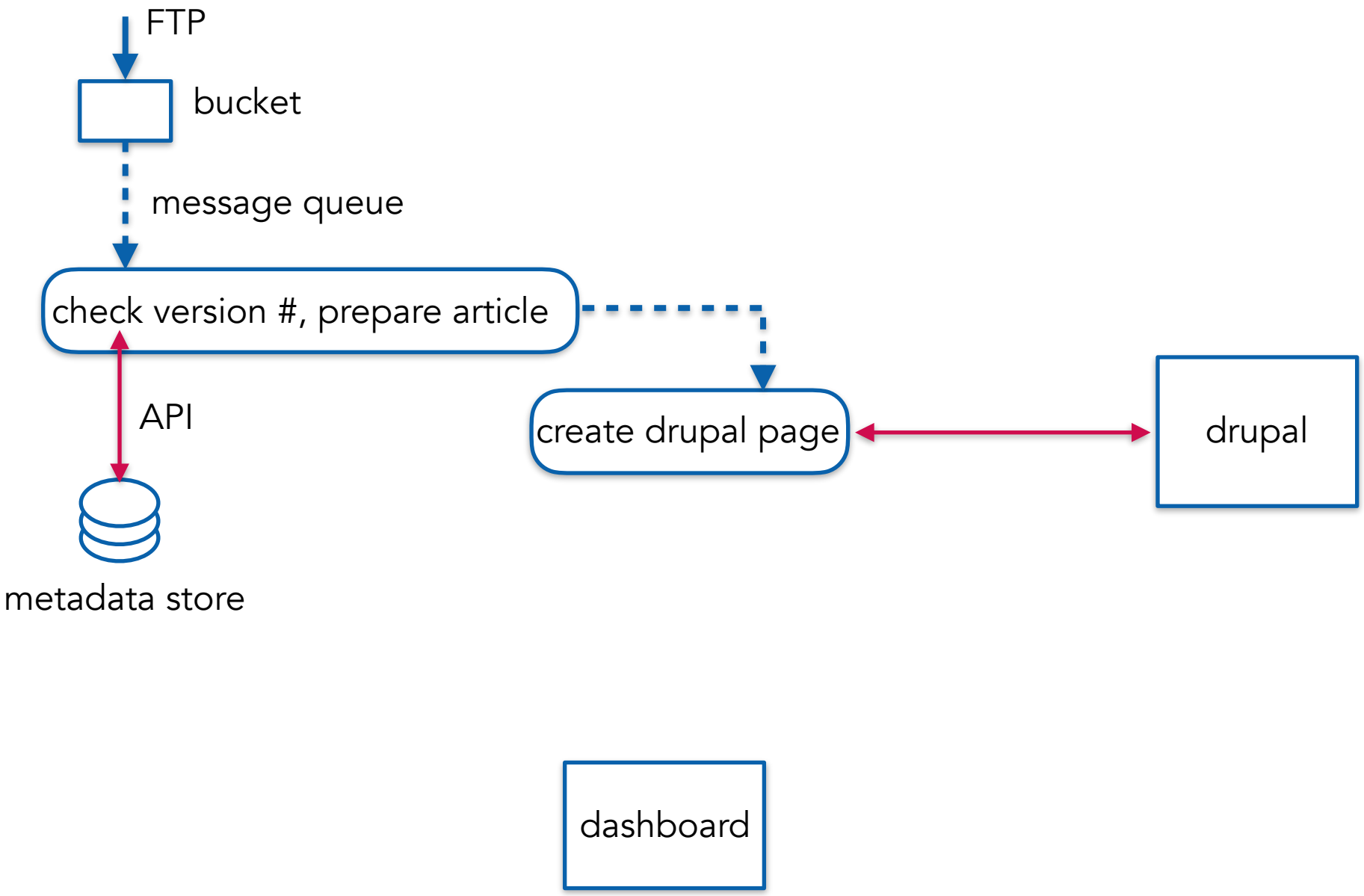

FTP
bucket

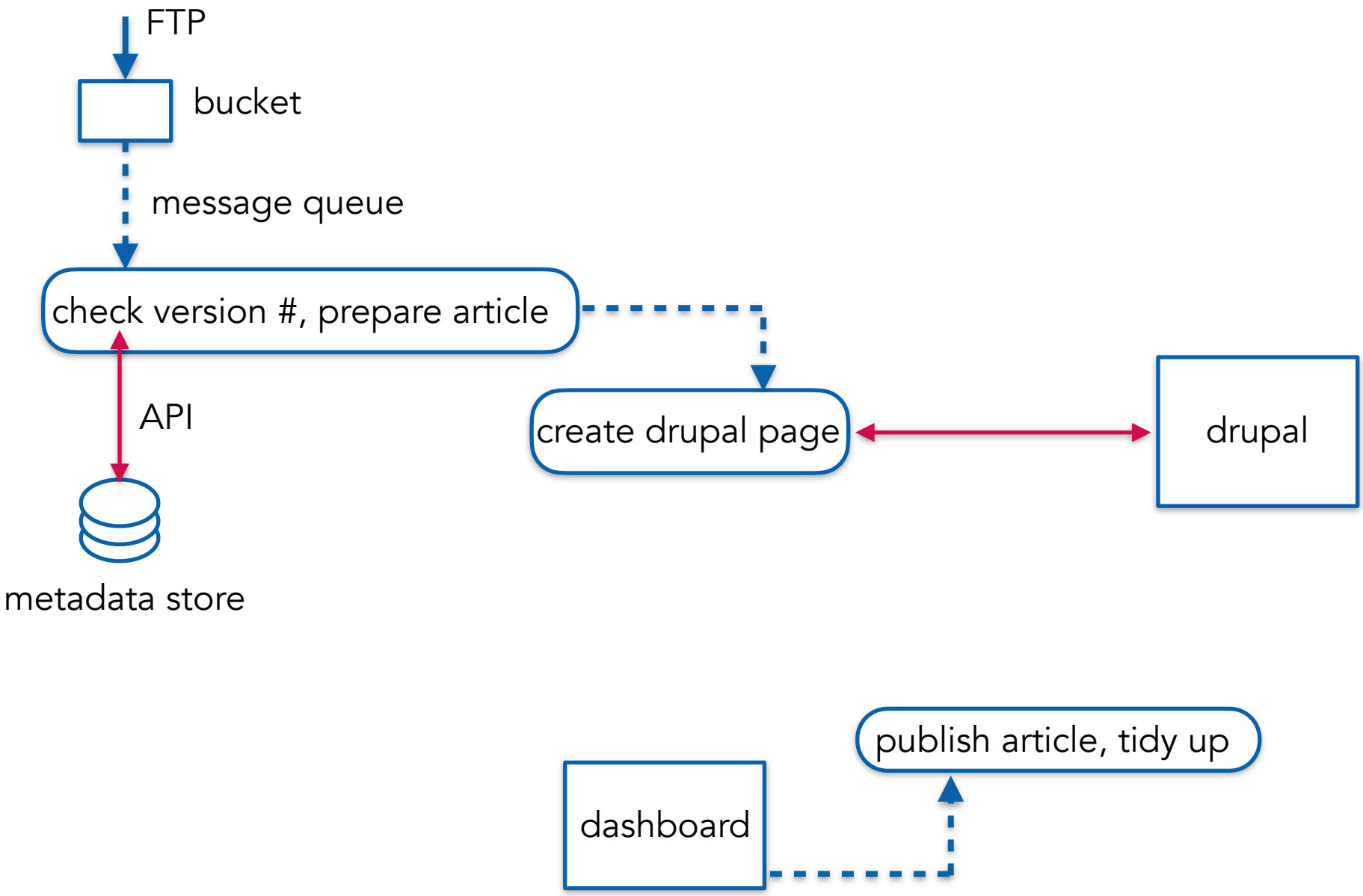

FTP
bucket
message queue
check version #, prepare article

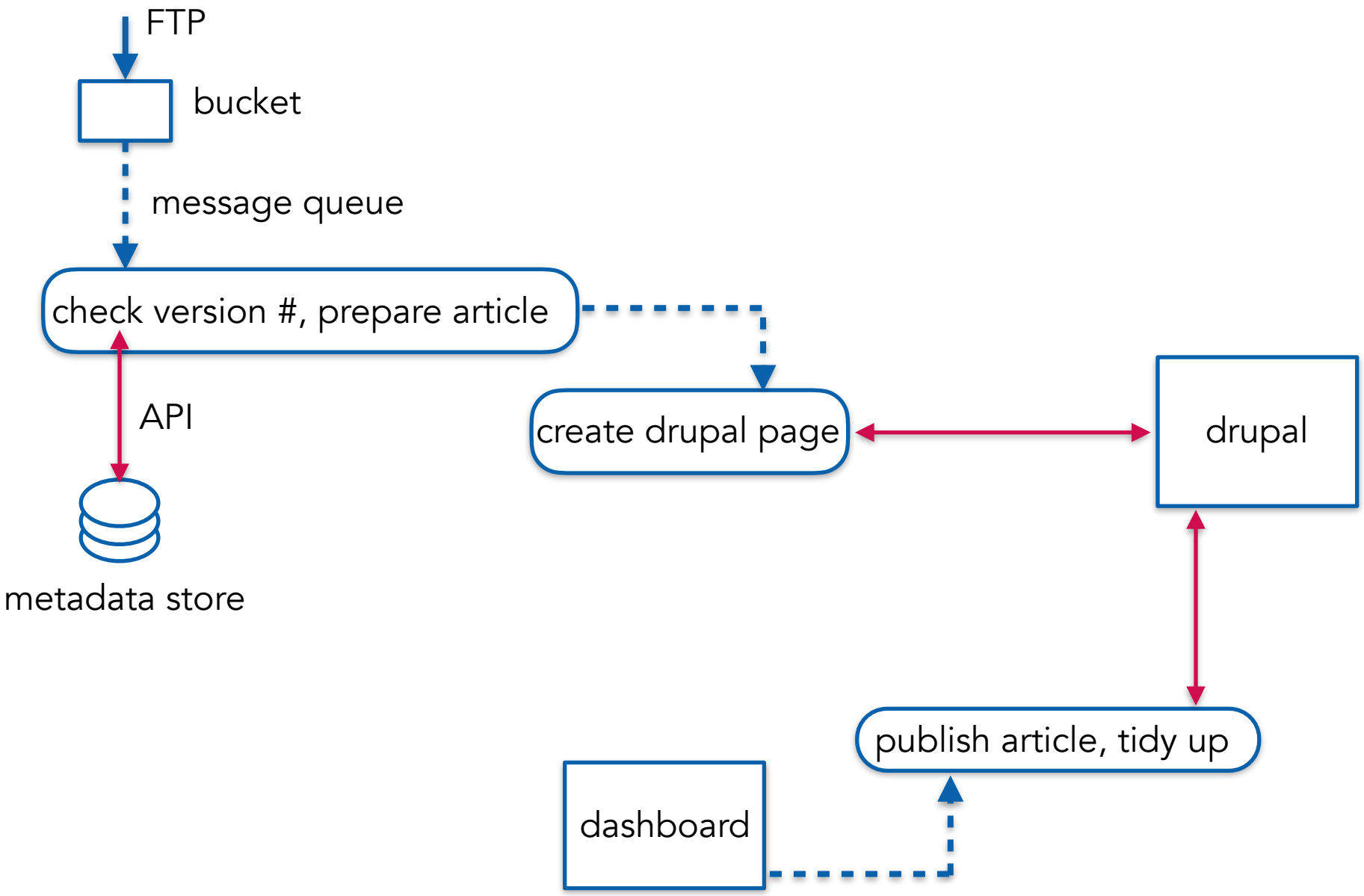

FTP
bucket
message queue
check version #, prepare article
API
metadata store

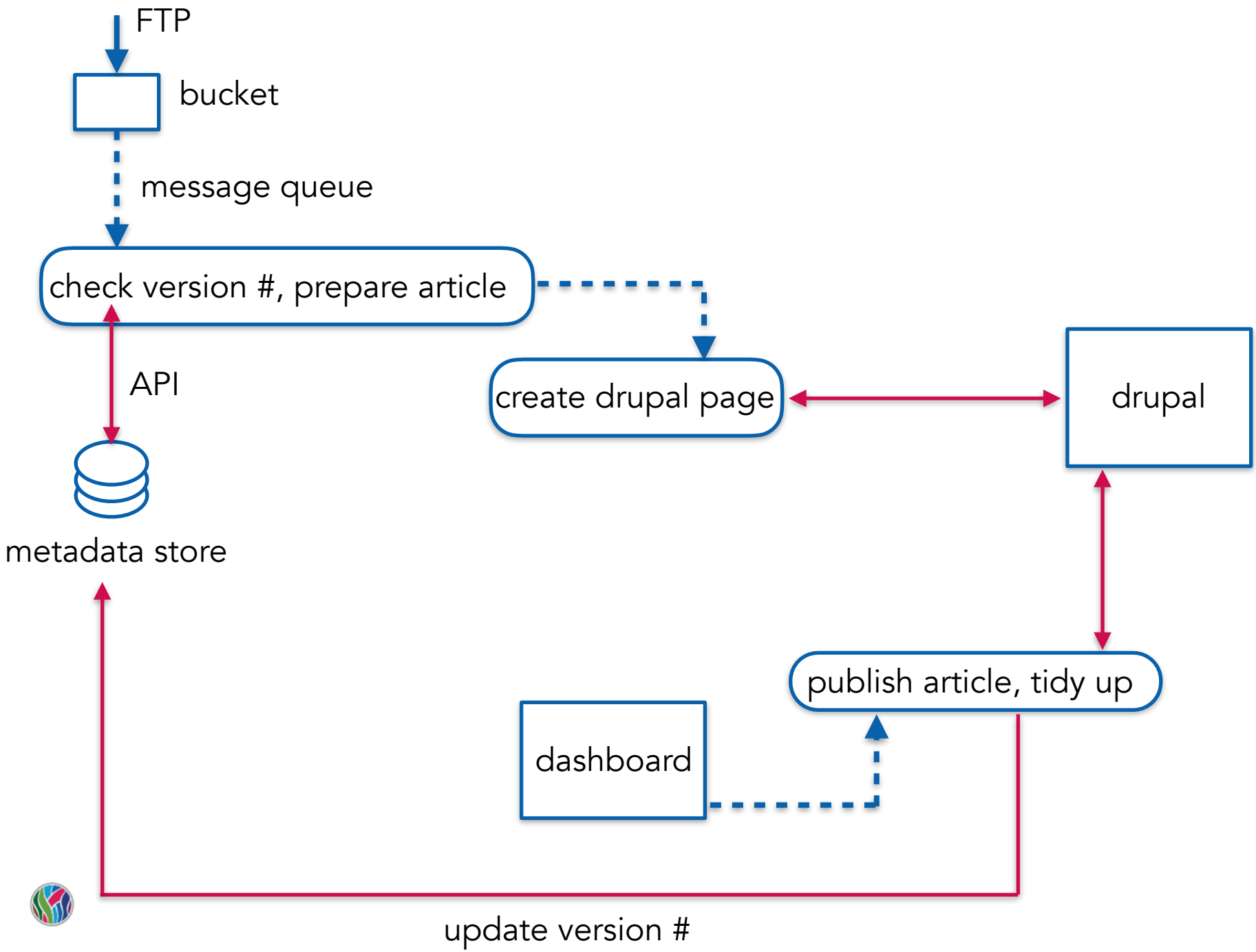

FTP
bucket
message queue
check version #, prepare article
API
create drupal page
metadata store

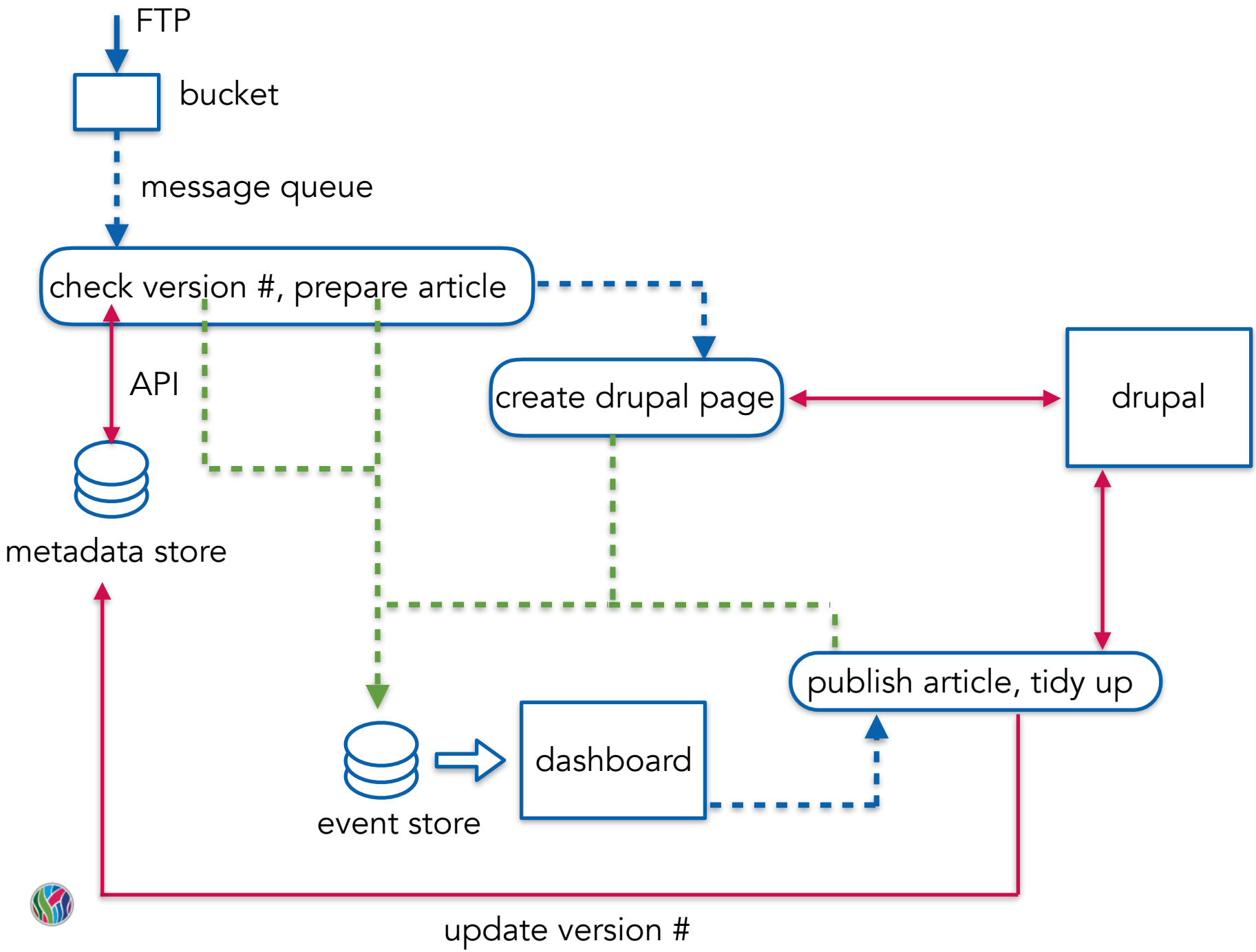

FTP
bucket
message queue
check version #, prepare article
API
create drupal page
drupal
metadata store

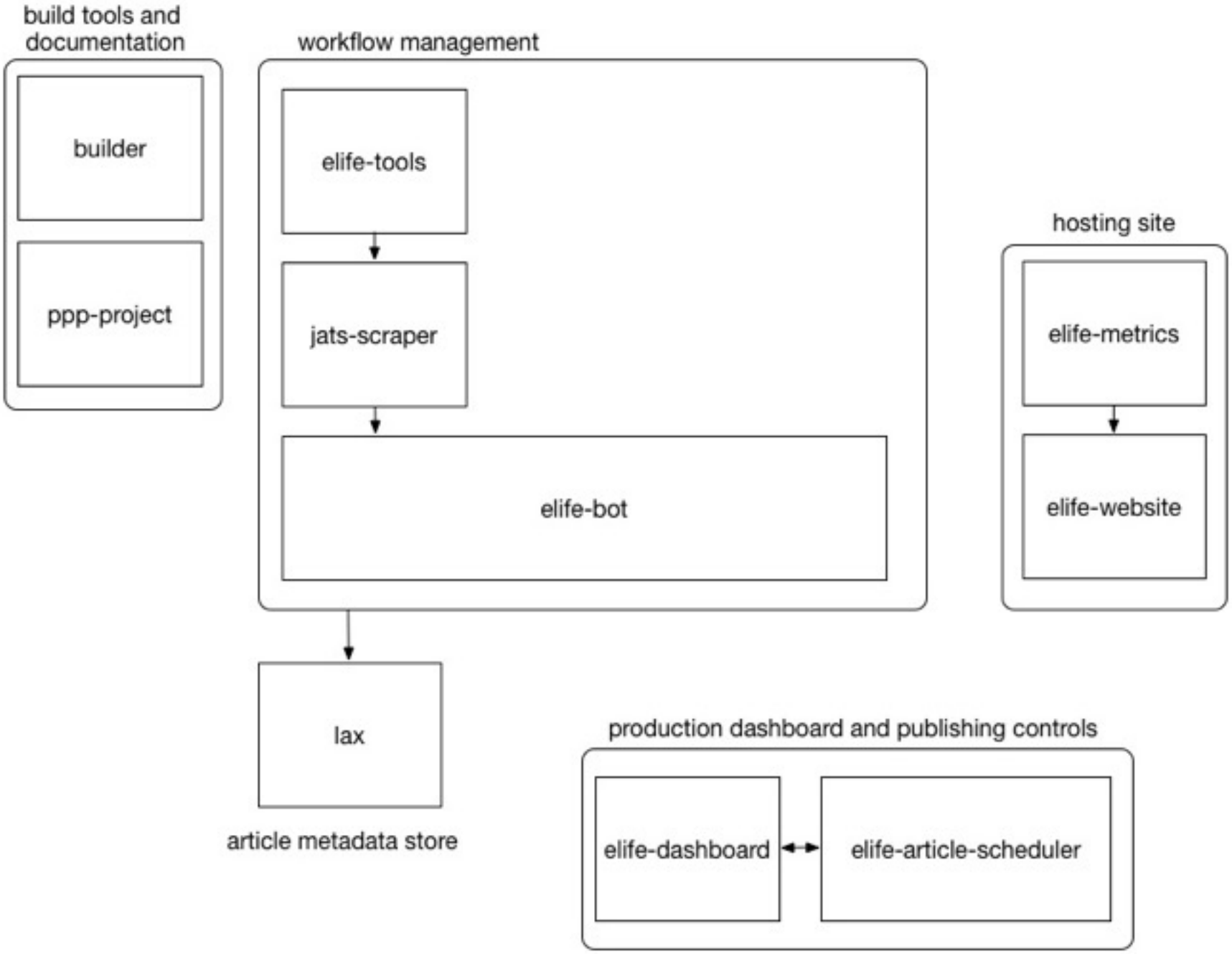

FTP
bucket
message queue
check version #, prepare article
API
create drupal page
drupal
metadata store
dashboard


FTP
bucket
message queue
check version #, prepare article
API
create drupal page
drupal
metadata store
publish article, tidy up
dashboard


FTP
bucket
message queue
check version #, prepare article
API
metadata store
create drupal page
drupal
publish article, tidy up
dashboard


FTP
bucket
message queue
check version #, prepare article
API
create drupal page
drupal
metadata store
publish article, tidy up
dashboard
update version #


FTP
bucket
message queue
check version #, prepare article
API
create drupal page
drupal
metadata store
publish article, tidy up
dashboard
event store
update version #


build tools and documentation
builder
ppp-project
workflow management
elife-tools
jats-scraper
elife-bot
hosting site
elife-metrics
elife-website
lax
article metadata store
production dashboard and publishing controls
elife-dashboard
elife-article-scheduler

- **Code Structure**

  - Anatomy of a workflow
  - Example of an activity
  - Triggering a new workflow
  - Sending an event to the event store

- **Demo**

  - https://continuum-test.ppp-dash.elifesciences.org/current
  - https://continuum-test.v2.elifesciences.org
  - http://ct-elife-production-final.s3.amazonaws.com/

• Amazon Web Services Dependencies

- Simple Workflow (SWF)

currently tightly coupled to SWF, but this is a cheap service.

- Simple Queueing Service (SQS)
- S3 & Bucket Notifications

SQS is fairly tightly coupled, but is just used for messaging between processes, could be replaced, e.g. RabbitMQ

S3 Buckets have FTP endpoints in front of them, only "AWS" feature that is used is bucket notification, couple be replaced by Cron

- Simple Mail Service
- Cloud formation

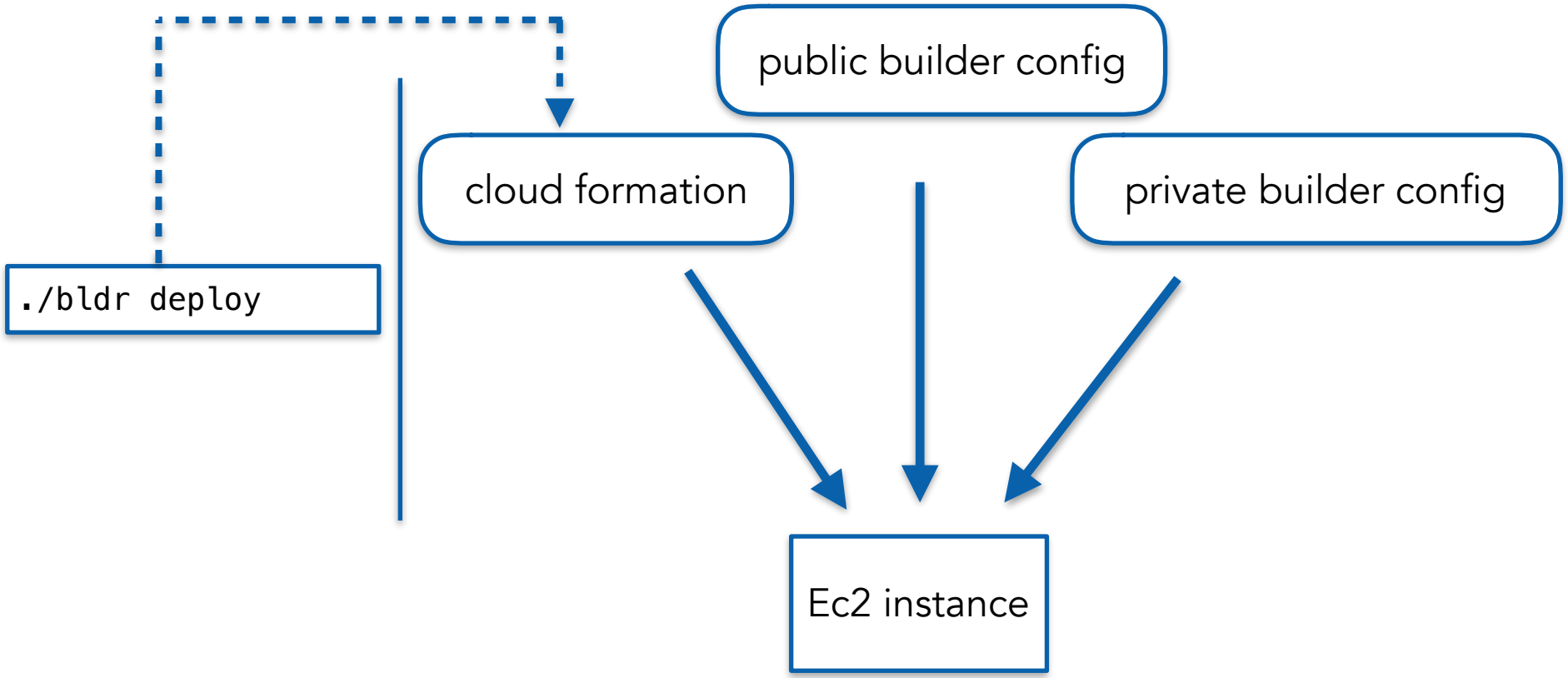
SMS and Cloudformation are for ease of use, but not needed for more core functions

- (EC2)

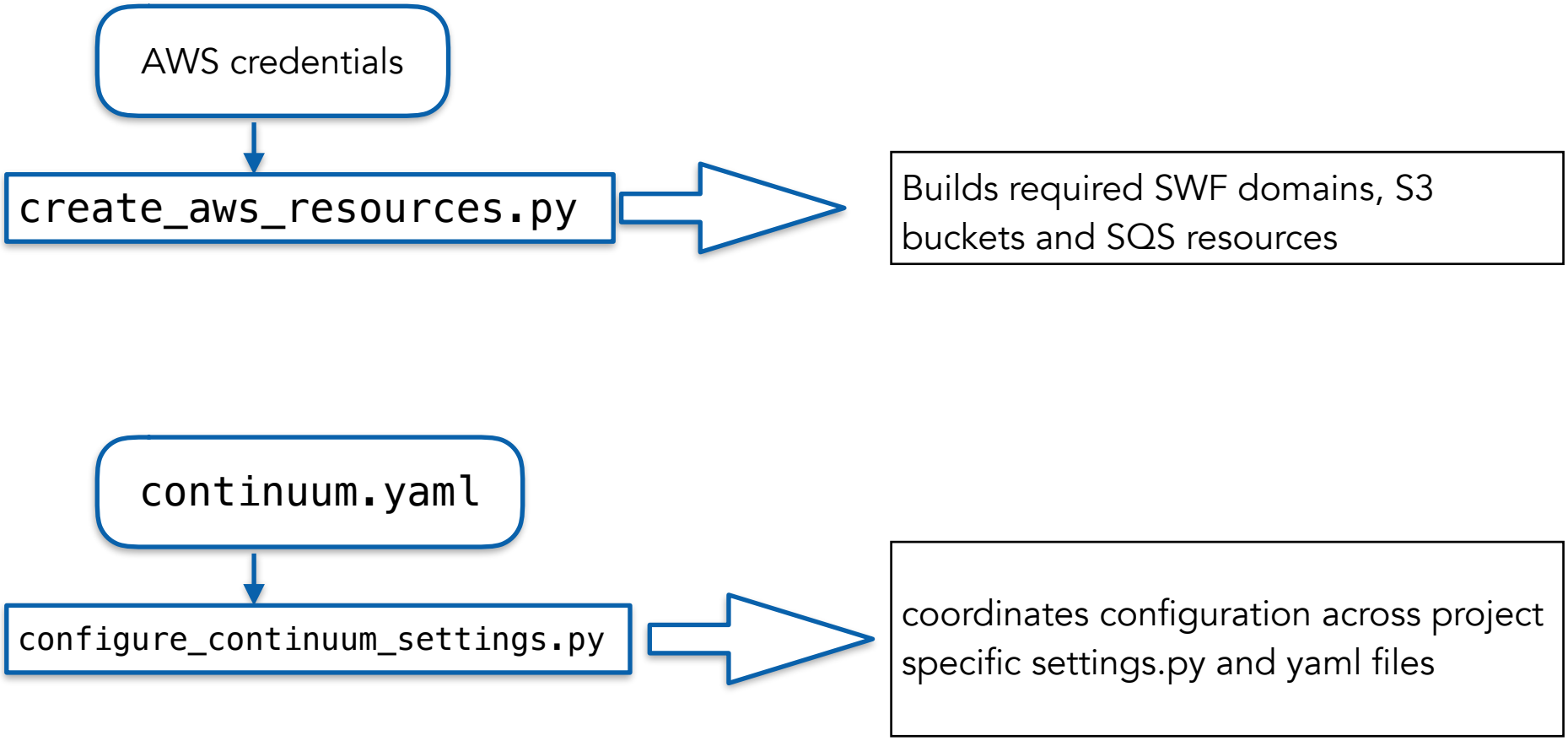
Processes run in Ec2, but could run anywhere, as SWF orchestrates workflows

- **Installation and Deployment**

https://github.com/elifesciences/builder uses http://saltstack.com



- **Installation and Deployment**



- **Next steps**

1. Continue to improve test coverage and integration with a CI server
2. Work towards the ability to make coordinated deployments
3. Work progressing on greater modularity of the back end
4. Get feedback from the community on features

- **Feature Requests and Feedback**

Code and documentation is available from

https://github.com/elifesciences/elife-continuum-documentation

Specific code / feature requests

https://github.com/elifesciences/elife-continuum-documentation/issues

General discussion / announcements

https://groups.google.com/forum/#!forum/elife-continuum-list

Thank you!

eLife is a non-profit organisation inspired by research funders and led by scientists

Q&A

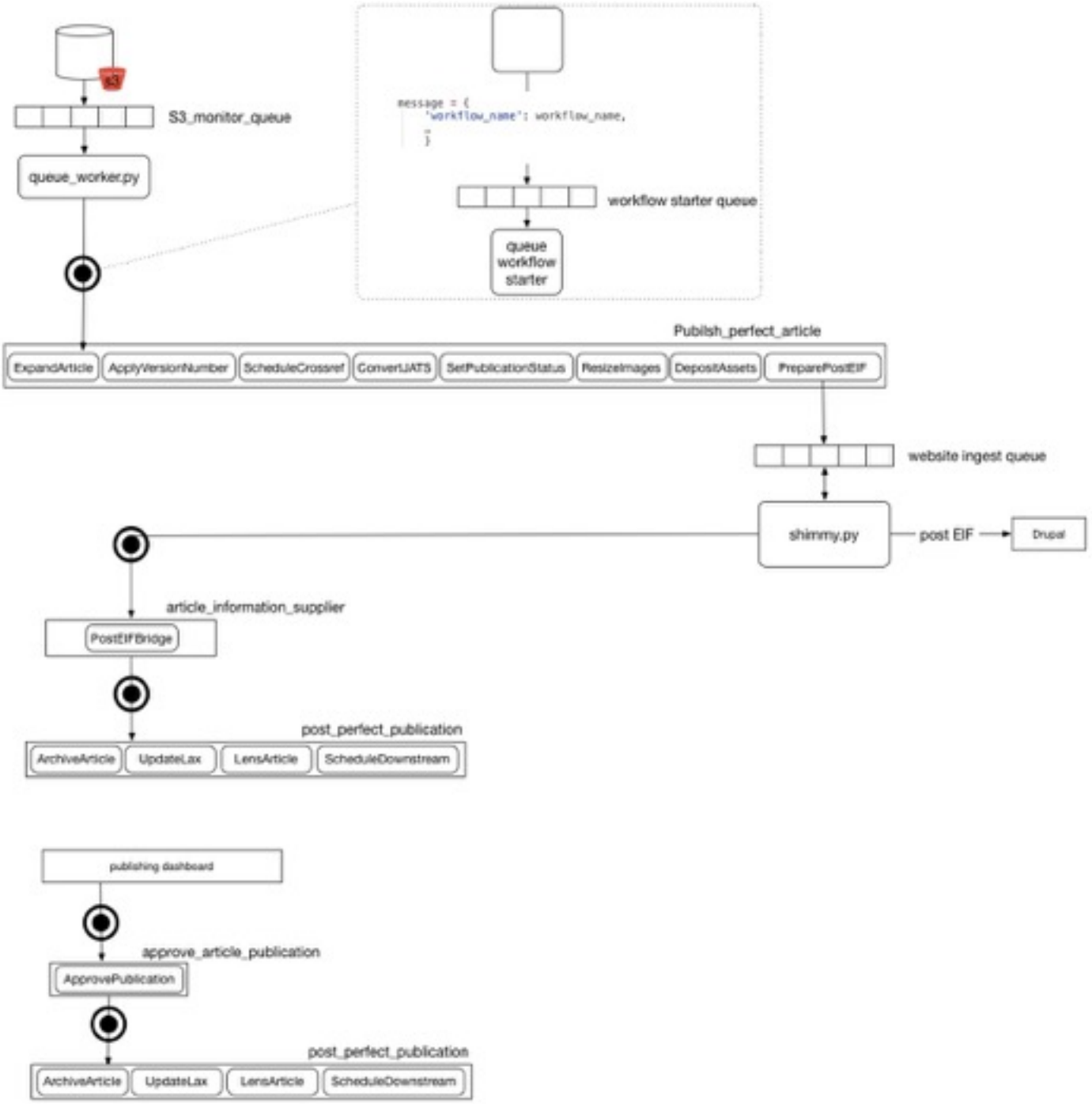
How to participate

1. Type in the question box on the right-hand side of your screen
2. Organisers will read out and answer the questions in order they are submitted.

Thanks for joining us

Please email further comments and questions to us at staff@elifesciences.org

- Appendix - supporting slides


S3_monitor_queue
queue_worker.py
message = {
'workflow_name': workflow_name,
}
workflow starter queue
queue workflow starter
Publish_perfect_article
ExpandArticle
ApplyVersionNumber
ScheduleCrossref
ConvertJATS
SetPublicationStatus
ResizeImages
DepositAssets
PreparePostEIF
website ingest queue
shimmy.py
post EIF
Drupal
article_information_supplier
PostEIFBridge
post_perfect_publication
ArchiveArticle
UpdateLax
LensArticle
ScheduleDownstream
publishing dashboard
approve_article_publication
ApprovePublication
post_perfect_publication
ArchiveArticle
UpdateLax
LensArticle
ScheduleDownstream

• supporting slide - main activities supported

ExpandArticle	unzips the content from the typesetter
ApplyVersionNumber	sets new version number for article
ScheduleCrossref	coordinates crossref deposit
ConvertJATS	XML -> JSON conversion for metadata
SetPublicationStatus	preset as published or private based on configuration
ResizeImages	convert images for web publishing
DepositAssets	populate the CDN
PreparePostEIF	populate Drupal with article
ArchiveArticle	on publication create an archive version of the article
LensArticle	generate eLife lens version of article
UpdateLax	update metadata store with new article version
ScheduleDownstream	prepare delivery to other places, e.g. PMC

- **Installation and Deployment - more details**

1. clone a copy of https://github.com/elifesciences/builder
2. create your own private version of https://github.com/elifesciences/builder-private-example on github
3. configure AWS and settings using https://github.com/elifesciences/elife-continuum-documentation/blob/master/elife-continuum-docs/create_aws_resources.py and https://github.com/elifesciences/elife-continuum-documentation/blob/master/elife-continuum-docs/configure_continuum_settings.py
4. launch instances of the required services using the`./bldr` command

Deployment remains a little brittle, it's not where we want it to be right now, but should be deployable early next week via builder.