Building and Deploying a Saas platform On Prem

A Digital Asset Management System as a Service

Christophe Vanlancker
@Carroarmato0

Slides by Michel van de Ven and Julien Pivotto

Christophe Vanlancker

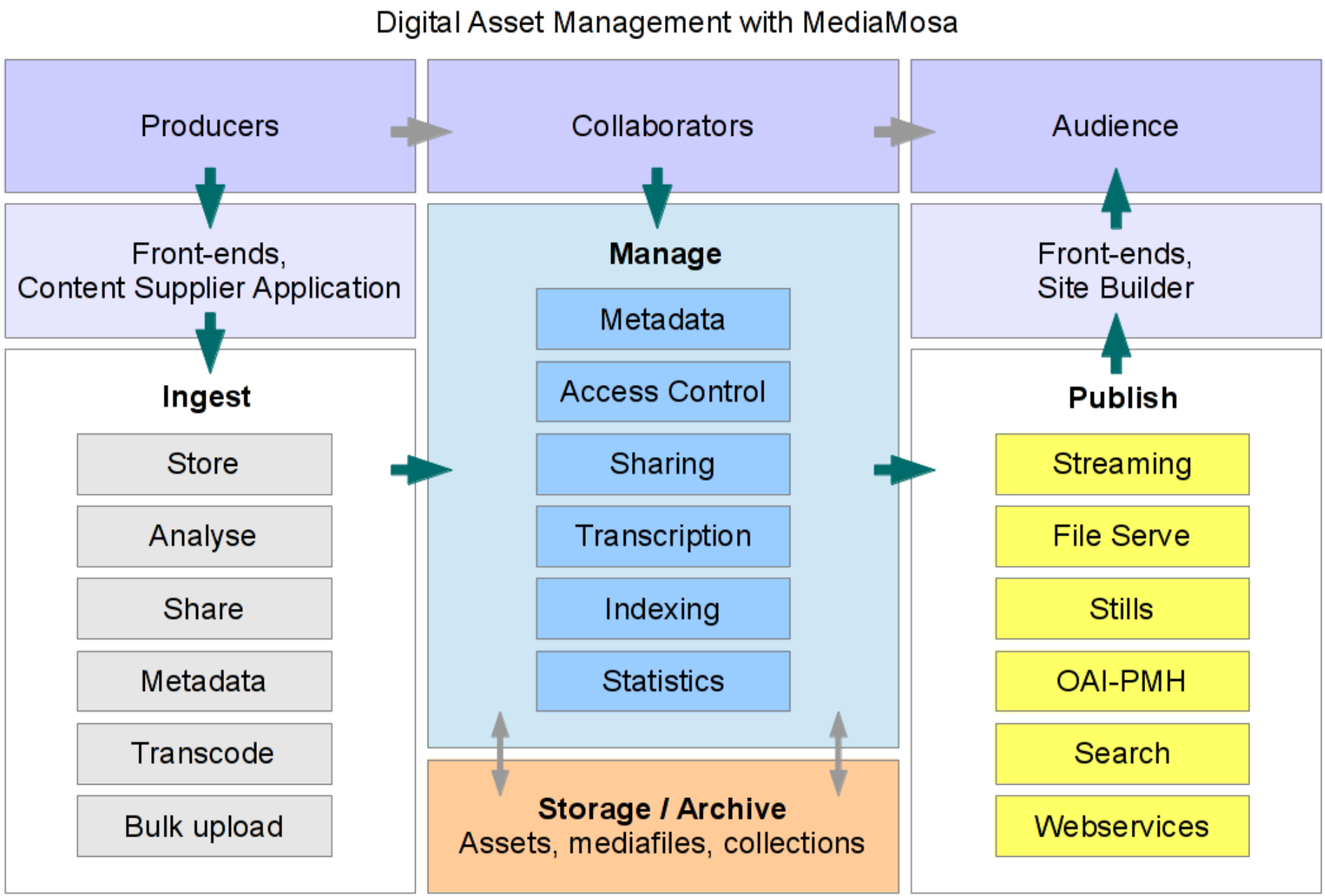
- Internal operations and consulting
- Mentor
- Kris couldn’t make it so I ‘s/Kris/Christophe/g’

Inuits

- Inuits is an Open Source company
  - We contribute back
- +70 people in 4 countries (BE, NL, UA, CZ)
- One language: English
- We offer
  - Consulting
    - Development
    - System Administration
  - and a niche Saas Platform

MediaMosa

- Drupal-based Digital Asset Management system
- Commissioned by SURFnet and Kennisnet
- Open Source (GPLv2), Open Standards
- Webservice oriented: REST
- Store & retrieve assets
- Manage metadata using open standards
  - Dublin Core, Qualified DC, IEEE/LOM, CZP
  - OAI-PMH, BagIt
- Transcode video, audio, images, PDF, stream content
- Users: Kennisnet, NIBG, UGent, UvA, TiU, RUG, Radboud, UOslo, Avans, PolitieAcademie, Acquia (NBC Sport), Cineca
- http://mediamosa.org


Digital Asset Management with MediaMosa
Producers
Collaborators
Audience
Front-ends,
Content Supplier Application
Manage
Metadata
Access Control
Sharing
Transcription
Indexing
Statistics
Front-ends,
Site Builder
Ingest
Store
Analyse
Share
Metadata
Transcode
Bulk upload
Storage / Archive
Assets, mediafiles, collections
Publish
Streaming
File Serve
Stills
OAI-PMH
Search
Webservices

=

MediaMosa as a Service

MediaSalsa infrastructure (simplified)

- For each environment (DTAP)
  - Backends: Core service (MediaMosa)
  - Frontends: Optional
  - Web servers
  - Database servers
  - Solr servers
  - Transcoding servers

Ideal world vs budget and reality
→ pragmatic approach

Culture

Automation

Measurement

Sharing

Puppet

Puppet automates all the things

→ mcollective orchestrates all the things

CD

- Continuous Delivery vs Continuous Deployment
  - “Continuous Delivery doesn't mean every change is deployed to production ASAP. It means every change is proven to be deployable at any time” (@ccaum)
- Puppet code
  - Deployed to dev environment
  - Same puppet code for each environment
  - User-triggered deployments to UAT & Prod
  - Feature flags in Puppet code per environment (switchable architecture)
- Application code
  - Continuous integration in dev
  - User-triggered deployments to UAT
  - Deployment to prod is a business decision

Testing

- Developers test a lot, but
  - The tests don’t work
  - It works on my machine™
  - Wrong platform
  - Wrong PHP version

  Fixed now, thanks to Jenkins!

  and Vagrant...

Version Control

- Git
- Code is under revision control
  - Prefer small commits
  - Local features branches
- Infrastructure as code → git / hiera

Using OS packaging system

- Consistency, security, dependencies
- Uniquely identify where files are coming from
- Source repo may not be reachable
- Little overhead when you automate
- Configuration does not belong in a package

Pipelines

- A collection of jobs
- Run in sequence
- Start on checkout, end on deployment
- From the developers' side:

  → Git push

  ← Mail with changes + link to deploy

Pipelines steps

- Checkout
- Syntax: php -l
- Style: Drupal Coder
- Package: FPM
- Deploy to dev environment: mcollective
- Tests in dev environment: drush run-tests
- Publish package and promote: mcollective

Same Pipelines, Tools, Patterns
are used by both devs and ops

Culture
Automation
Measurement
Sharing

Logstash

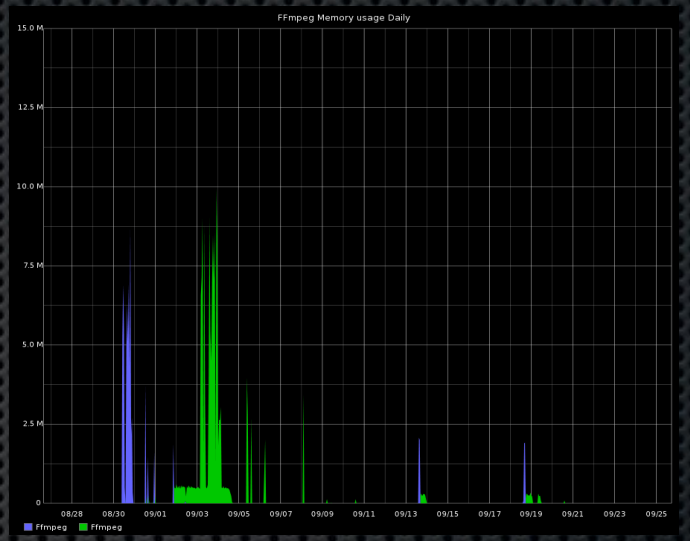
- Collect all the logs
  - Drupal logs
  - Apache logs
  - Deployment logs
  - System logs
- Interpret, filter and correlate them
- Logstash, ElasticSearch, Kibana, statsd, Graphite

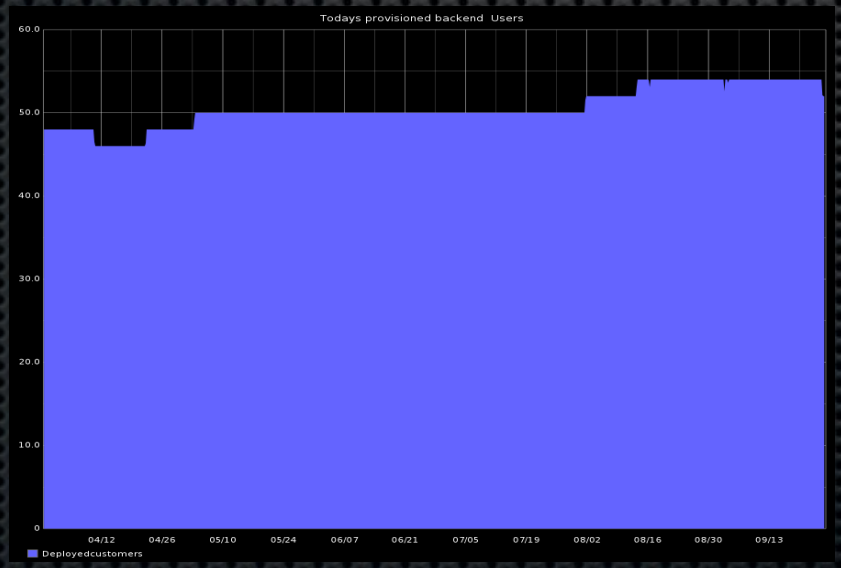
Icinga

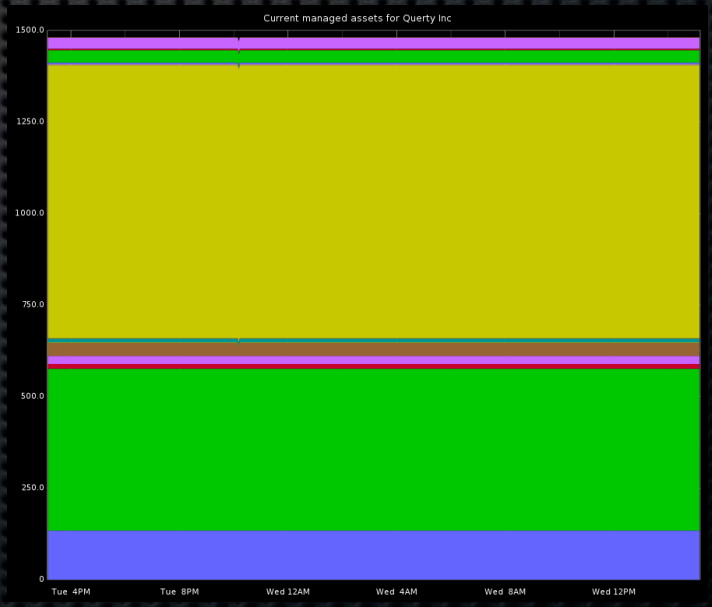
- Monitor everything
  - vhosts
  - databases
  - cronjobs
  - unit test suites

Graphite + gdash

- Collectd
- Monitor platform usage
- FFmpeg usage
- Number of accounts
- Pipelined !







Culture
Automation
Measurement
Sharing

Open Source

- Mediamosa is Fully Open Source
- Lots of the PuppetCode to deploy it
- Our passwords etc aren't

MediaSalsa Deployments

- Initially
  - 1 instance Academic usage @SurfNet
  - 1 Instance Commercial DC for non-edu
- Today
  - 2 academic instances
  - 1 commercial Saas instance
  - 2 on prem deployments

Why multiple Deployments

- “Security”
  - Academic Customer wanted a private tenant for security and privacy
- Initial hardware investment done already
  - Public Tender , $customer bought huge amount of storage
  - Saas solution charges per TB
  - Asked for custom manual deployment
- CIO’s don’t believe in Cloud/SAAS (2017 !!!!)

Saas vs OnPrem

We have automated everything,

Infrastructure as Code , Pipeline as Code, Continuous delivery , so deploying this stack another time should be trivial !!

WRONG

Biased Automation

- Works in our infra , our constraints, our expectations
- We expect to have access to our infra
  - Puppet, monitoring, metrics, repos , jenkins

VM Provisioning

- Different Technologies
  - Open vs Proprietary
  - Guess which one is more problematic
- No access to Internal repositories
- Network topologies
- Having to ask to reboot a host
- Having to ask to grow a VM

Security

- IPSec links to all stacks
  - Our own network complexity has grown exponentially
- Our network = Trusted
- Their network = Hostile
  - Different approach in host vs network based firewalling
- User management
  - Only our accounts in our stack , our ldap
  - ~~They want accounts~~

Variants

- We don’t want exceptions
- They do want exceptions

- Old purchasing mentality
  - Custom Features
  - Additional Components
  - It’s “Their” stack

- Exceptions need to be codified in our infra

Continuous ~~Deployment~~ Delivery

- Deployment isn’t our decision anymore
- Back to fixed deployment windows :(
- Coordination with $customer on when to deploy
- Even for Security Fixes

- For every single instance

Ideal world vs budget and reality
→ pragmatic approach

Extreme Cost Difference

- The effort to run 5 stacks in your own infrastructure within your team is smaller than running 1 additional stack on prem at a customer
- Your pragmatic approach does not fit their infrastructure
- You will need to implement features (security/ storage support) that you do not need for your SAAS platform.

It could have been worse

- We are an Open Source company
- All of our Choices are Open Source by default
  - We could deploy full stacks On Prem
  - Including metrics, log analytics and monitoring
  - We had no external dependencies
  - No additional license costs