

Lesson: Data flow in the CLAW



March 15, 2016

(My name is Diego Pino Navarro
I work at metro.org)

The CLAW is Islandora for Fedora 4

Let's get serious with some diagrams

Simple stack diagram



OK, we failed with serious
We need doodles

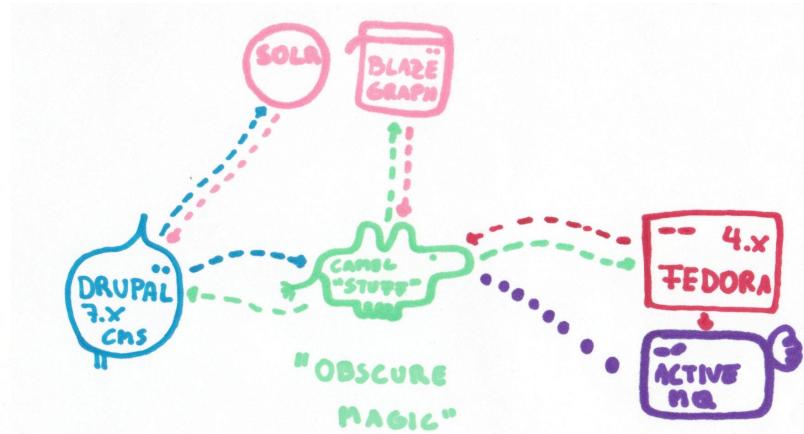
The usual suspects: a typical Claw environment



Basic interconnection (zoom level 0)



Where is ISLANDORA CLAW?



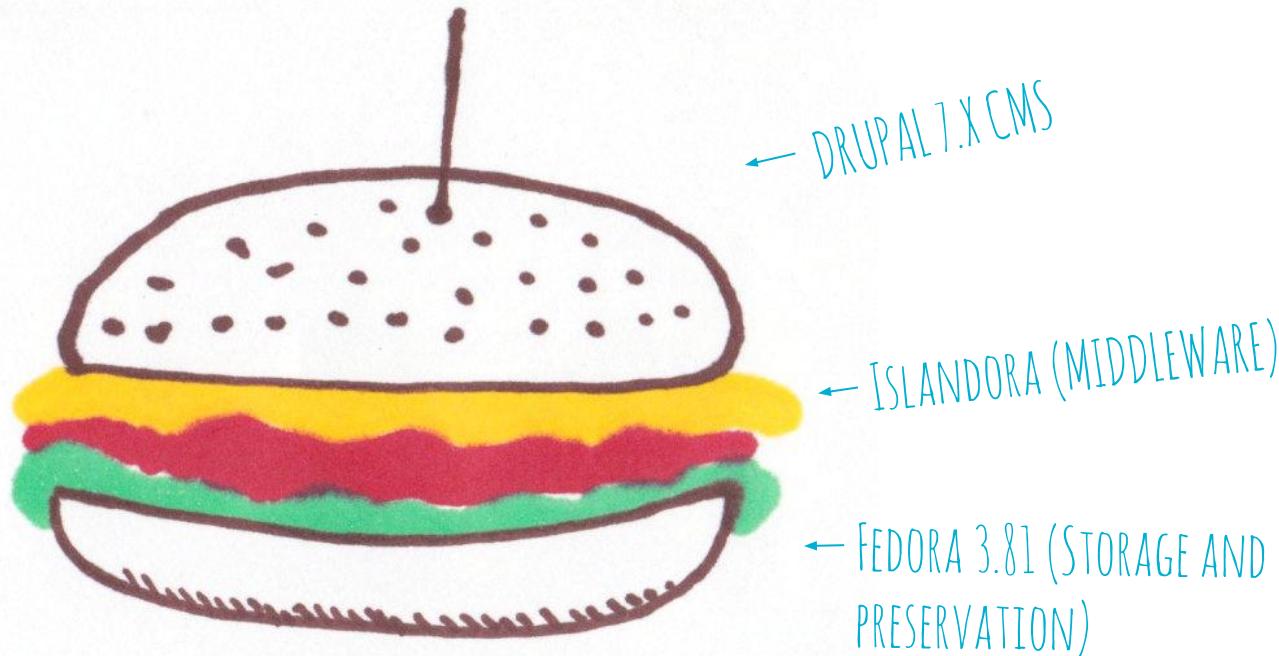
PLEASE DON'T TELL ME IT IS INSIDE THE "OBSCURE MAGIC" PART

Islandora-CLAW is a bit everywhere
(and among us)



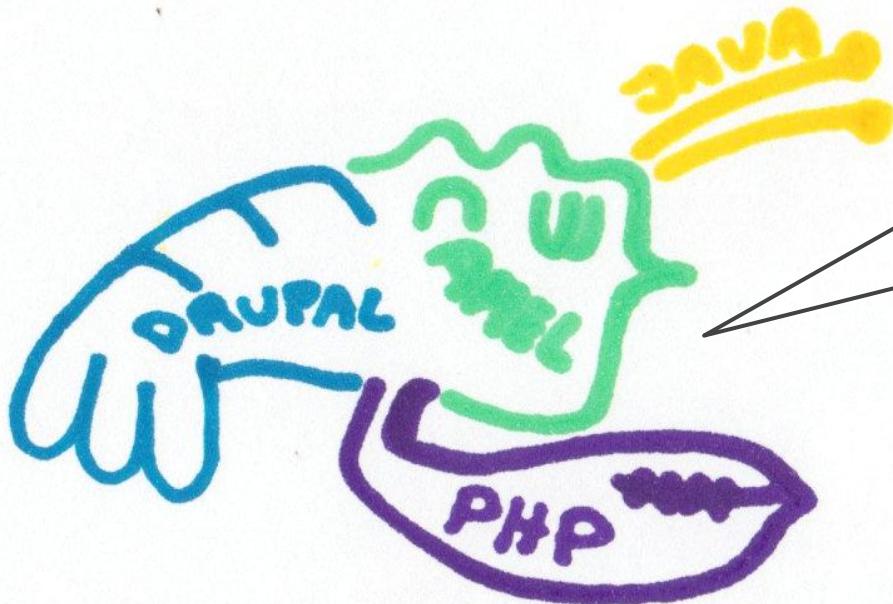
What we know about current Islandora (7.x-1.7)

<https://github.com/Islandora/Islandora>



What we should know about Islandora-CLAW

<https://github.com/islandora-claw/claw>



I MAKE USE OF EXISTING TOOLS
INTEGRATING MYSELF INTO
EACH ONE. I'M NOT A LAYER, I
PLAY ROLES IN MULTIPLE ONES.

Quick Facts about Islandora-CLAW



- Makes use of Drupal CMS completely
- Islandora Solution Packs are Real Drupal Modules
- Resources Live in Drupal and in Fedora 4 (same meaning, different representations)
- There is no “Source Master”.
 - Bi-directional Sync between Drupal and Fedora
- Exposes endpoints everywhere (RESTfull API)
- It’s written in PHP(variety of frameworks), JAVA, Spring, Blueprint XML
- It deals with data in a very Async fashion
- It’s a stack of moving pieces that depend on each other

Human Facts about Islandora-CLAW



- We are small but committed group of developers (5 Humans)
- Nick Ruest is our Project Director
- Dani Lamb is our Tech-lead
- We are committed to make this a good software
- Resolving one problem at a time
- We have big plans (Drupal 8, micro-services, etc)
- We are open to discuss and modify things

AND WE MEET EACH WEEK (YOU ARE INVITED BECAUSE WE NEED YOU)

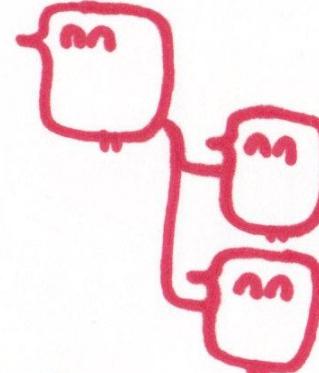
Data definition @ Islandora-CLAW



DRUPAL
CONTENT



FEDORA 4
RDF
RESOURCES



ISLANDORA GENERATES REAL DRUPAL
CONTENT.

ISLANDORA CRUDS REAL FEDORA 4 RESOURCES

Drupal Content the Islandora-CLAW way



DRUPAL
CONTENT

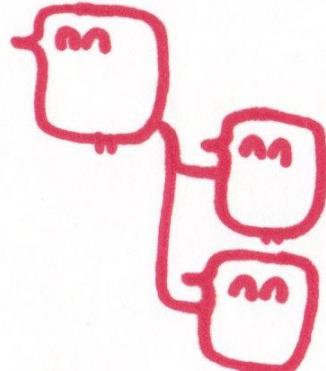


- DRUPAL provides Content types and entity types
- We extend the Drupal **NODE** type to make a **BUNDLE**
- Our **BUNDLES** include RDF fields (properties)
- Other fields too (configurable)
- We integrate **UUID** (it's back!)
- This whole content (**Entity**) lives in DRUPAL (MYSQL)
- Versionable as a whole
- We get an **URL**
 - <http://somedomain.com/node1>
(or UUID)

Drupal Content the Islandora-CLAW way

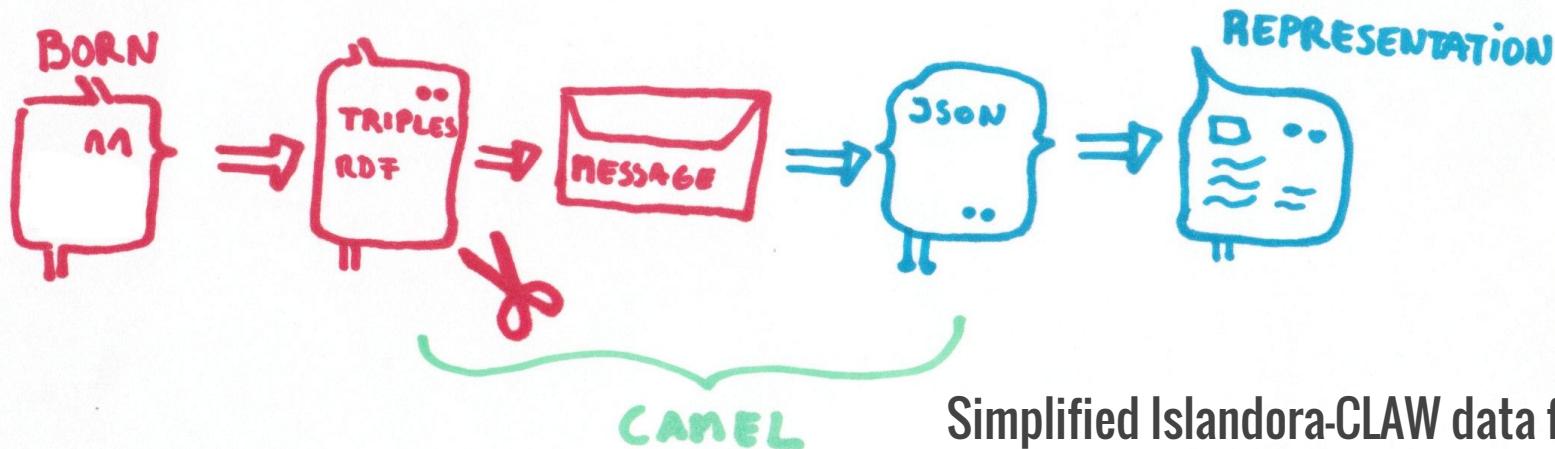
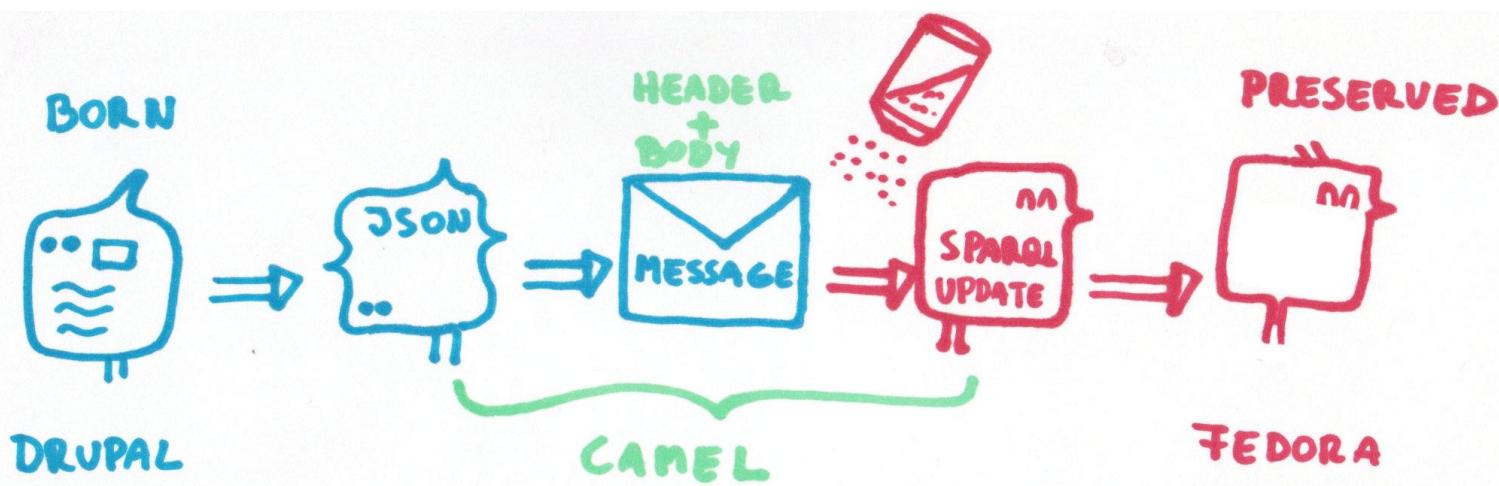


FEDORA 4
RDF
RESOURCES

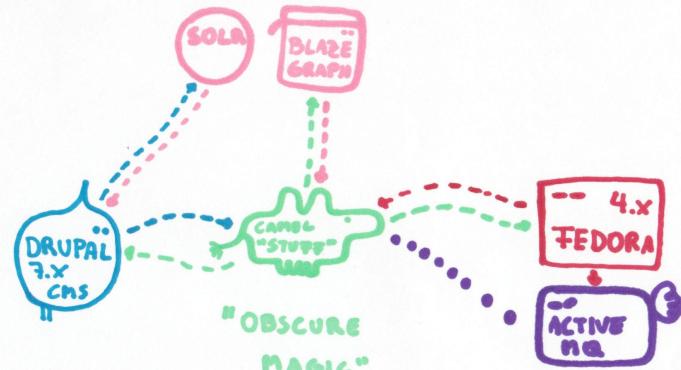
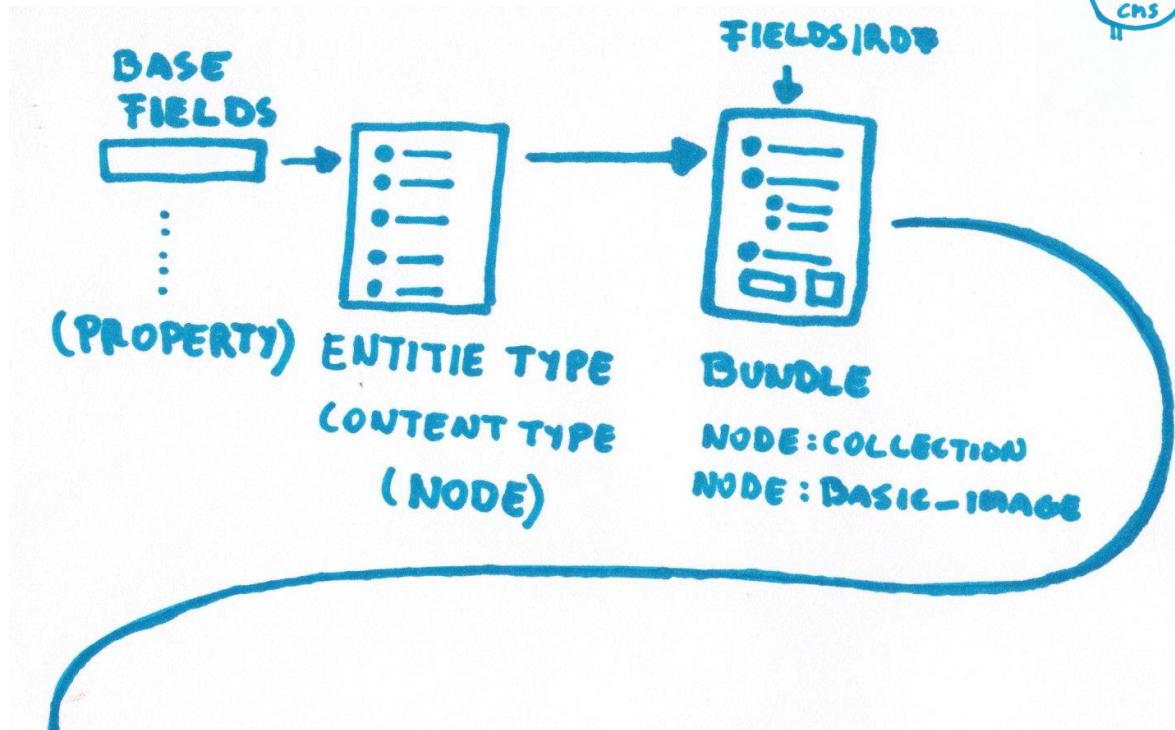


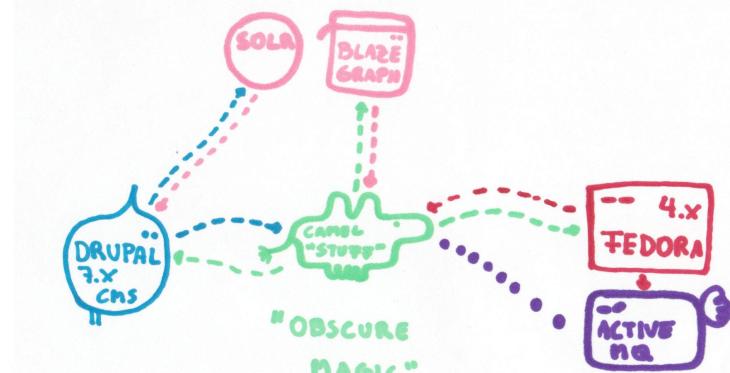
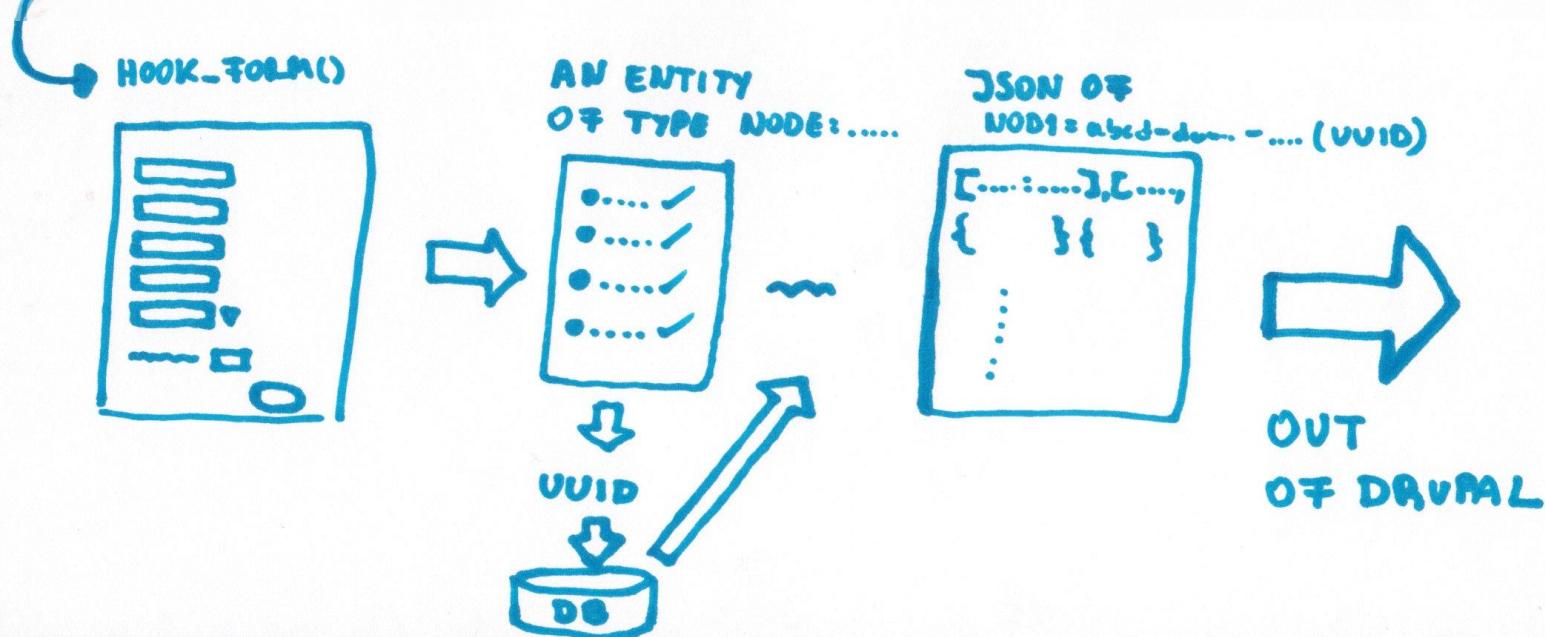
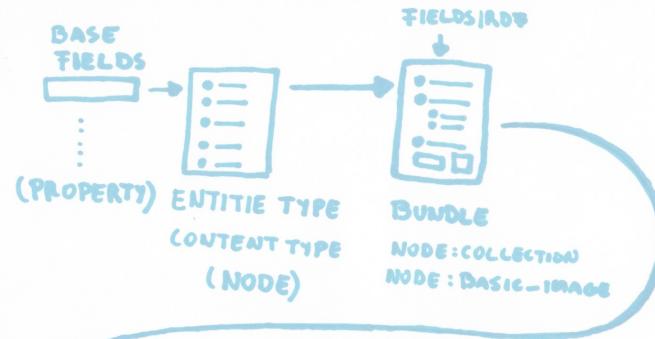
- Has rdf:types + properties
- It's a graph of resources
- LDP containers
- We get PATHS as ID
- Lives in Fedora 4
- Each resource can be Versioned

So we need to transform back and forth
(Welcome to the data flow)

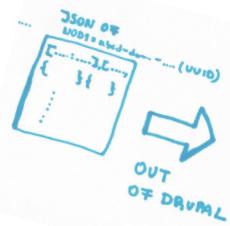


Same in finer detail: Let's Ingest!





CAMEL (zoom+1)



POST

JSON OF
ENTITY



NODE TO SPARQL

CREATE:...

MESSAGE

ISLANDORA
COMMANDS

FLERPO
CAMEL COMP.

(PHP)

(JAVA)

POST

OUT
OF CAMEL
(SPARQL UPDATE)

 POST
OUT
OF CAMEL
(SPARQL UPDATE)

POST



(APPLICATION)
SPARQL-UPDATE)

(RDF VERSION
OF DRUPAL
ENTITY)

FEDORA
4.X

ADDS REPO
TRIPLES
+ PATH

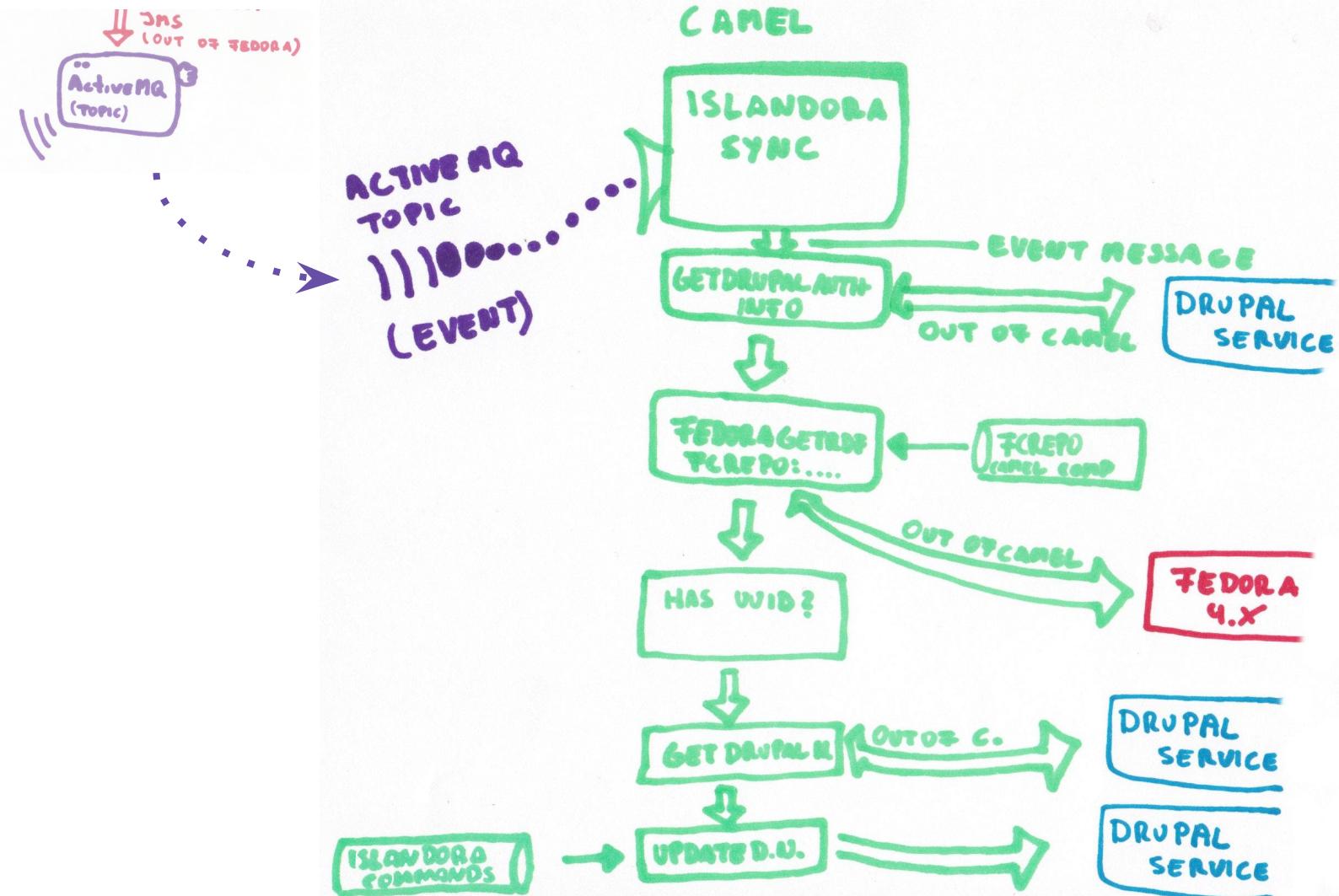
NODE-ADDED

MODESHAPe

EVENT

JMS
(OUT OF FEDORA)

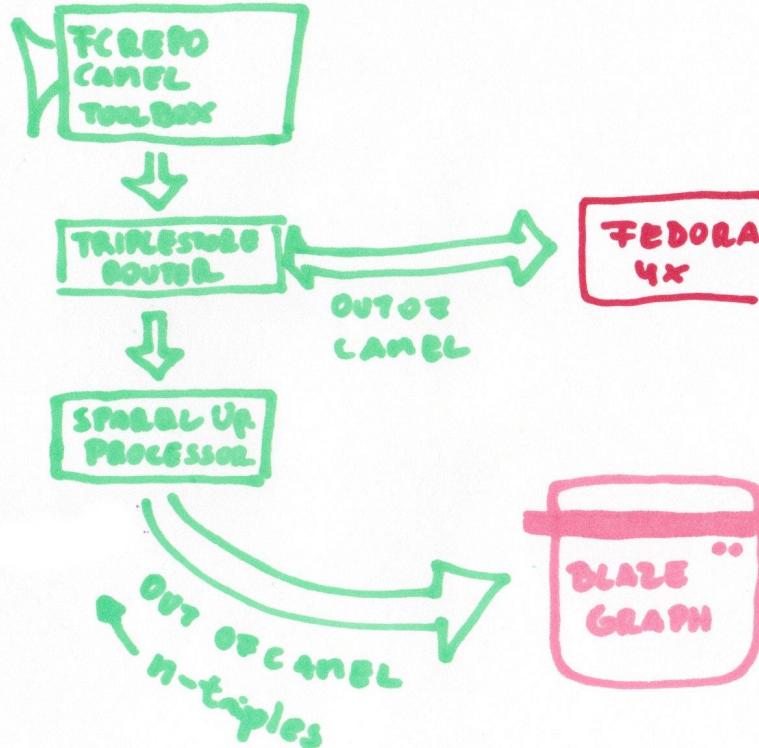
ActiveMQ
(TOPIC)



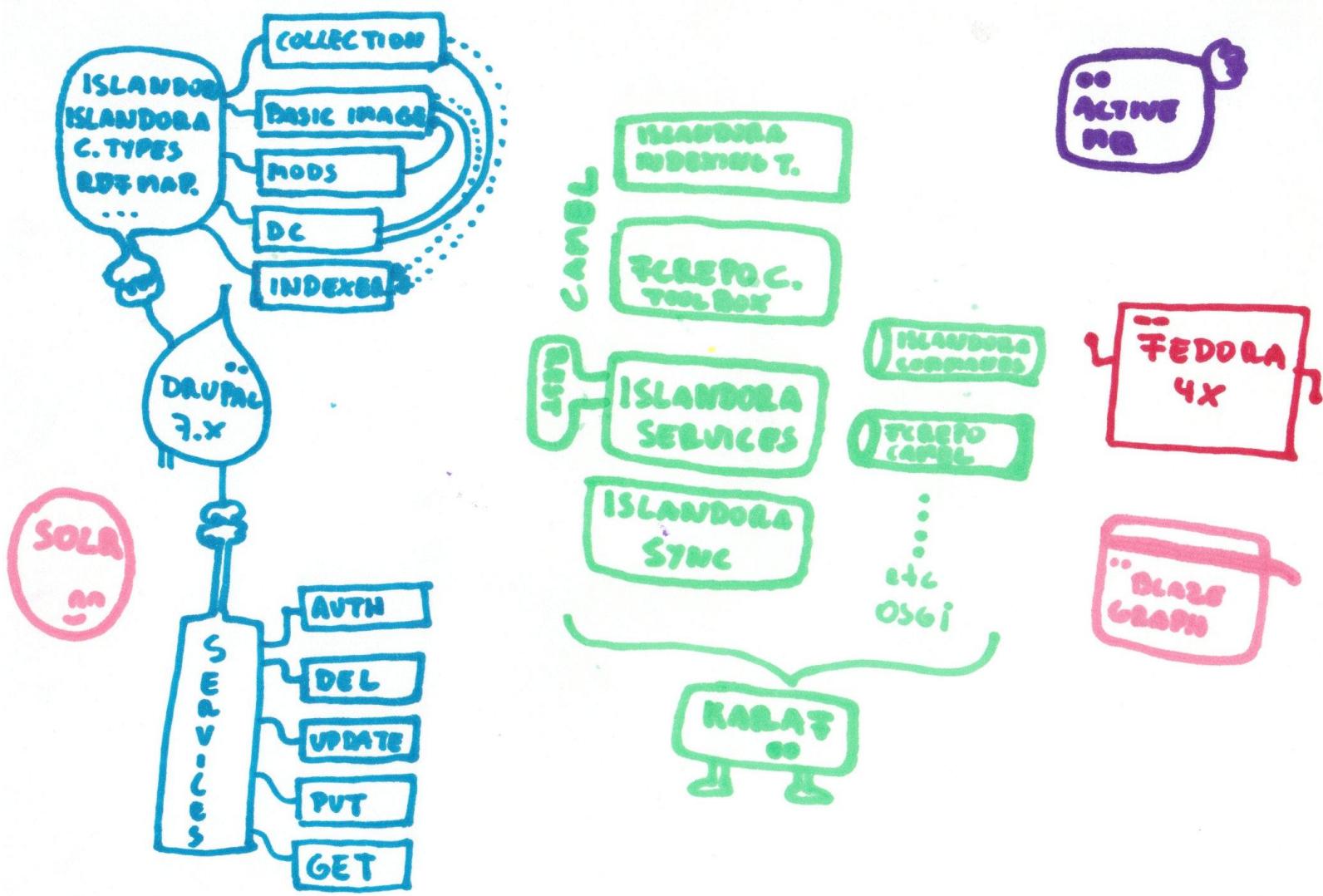
... at the same time

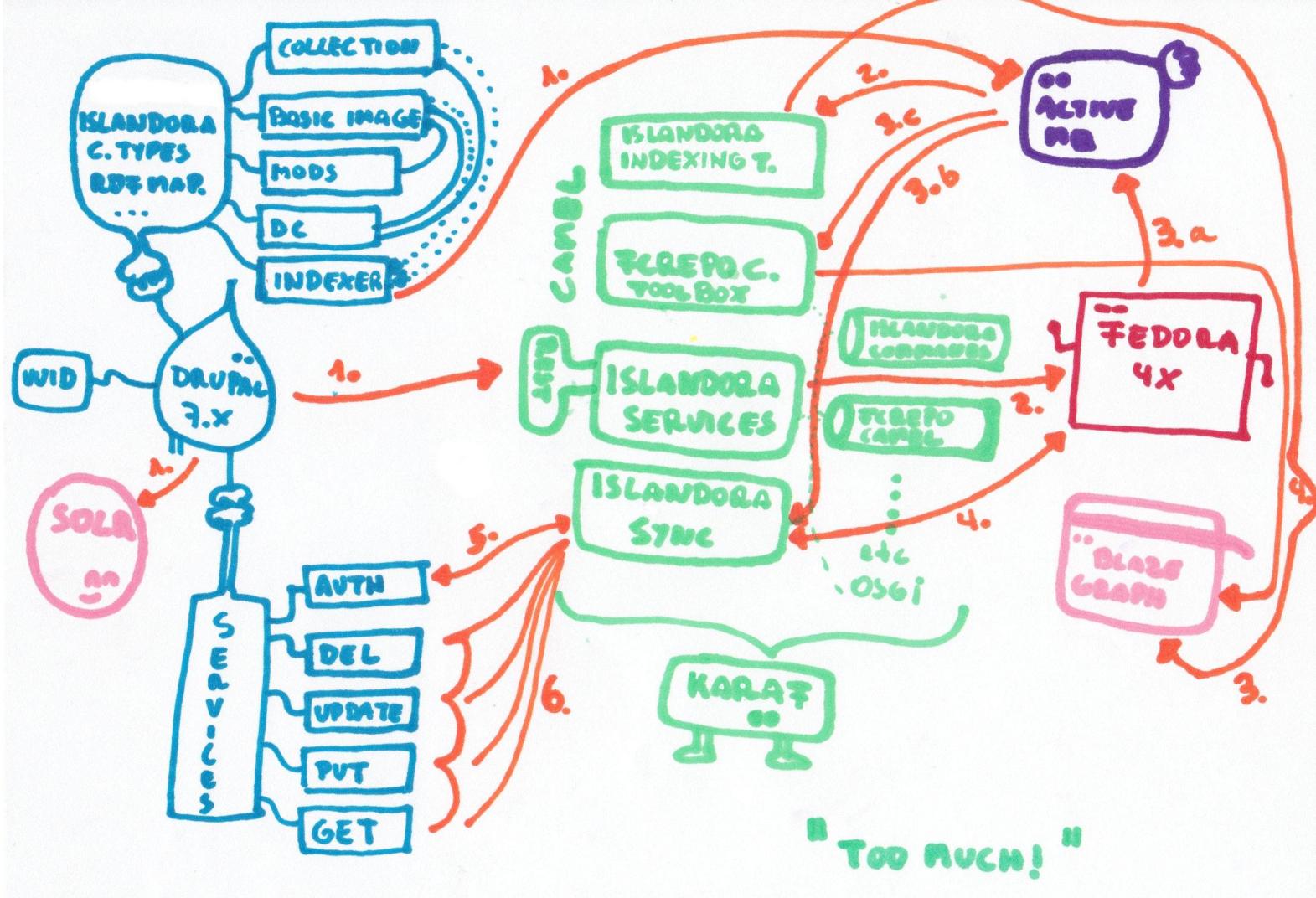
CAMEL

ACTIVE MQ
TOPIC
))))...
(EVENT)



This is so complicated
(Welcome to the data flow)





Next Session:
CLAW Hands on Drupal side