



DRUPALCON

DENVER

MARCH 19-23, 2012

Collaborative Publishing for Every Device

Drupalize Your Data: Use Entities!

Presented by Wolfgang Ziegler // fago

Wolfgang Ziegler // fago

- from Vienna, Austria
- studied at the TU Vienna
- Drupal since 2005.

wolfgangziegler.net

twitter.com/the_real_fago

[gplus.to/fago](https://plus.google.com/u/0/fago)

Outline

- Introduction
- Interact with entities
- Providing a new entity type
- Expose non-DB entities
- Outlook

Drupal 7: Fields everywhere!

Bundles

Entity type



Node

Bundle



Node type

Entity type



Taxonomy term

Bundle



Vocabulary

Entity type



User

Bundle



{}

Which entities are there?

- Core
- Modules:
 - Drupal commerce
 - Organic groups
 - Profile2
 - Heartbeat and Message
 - File entity

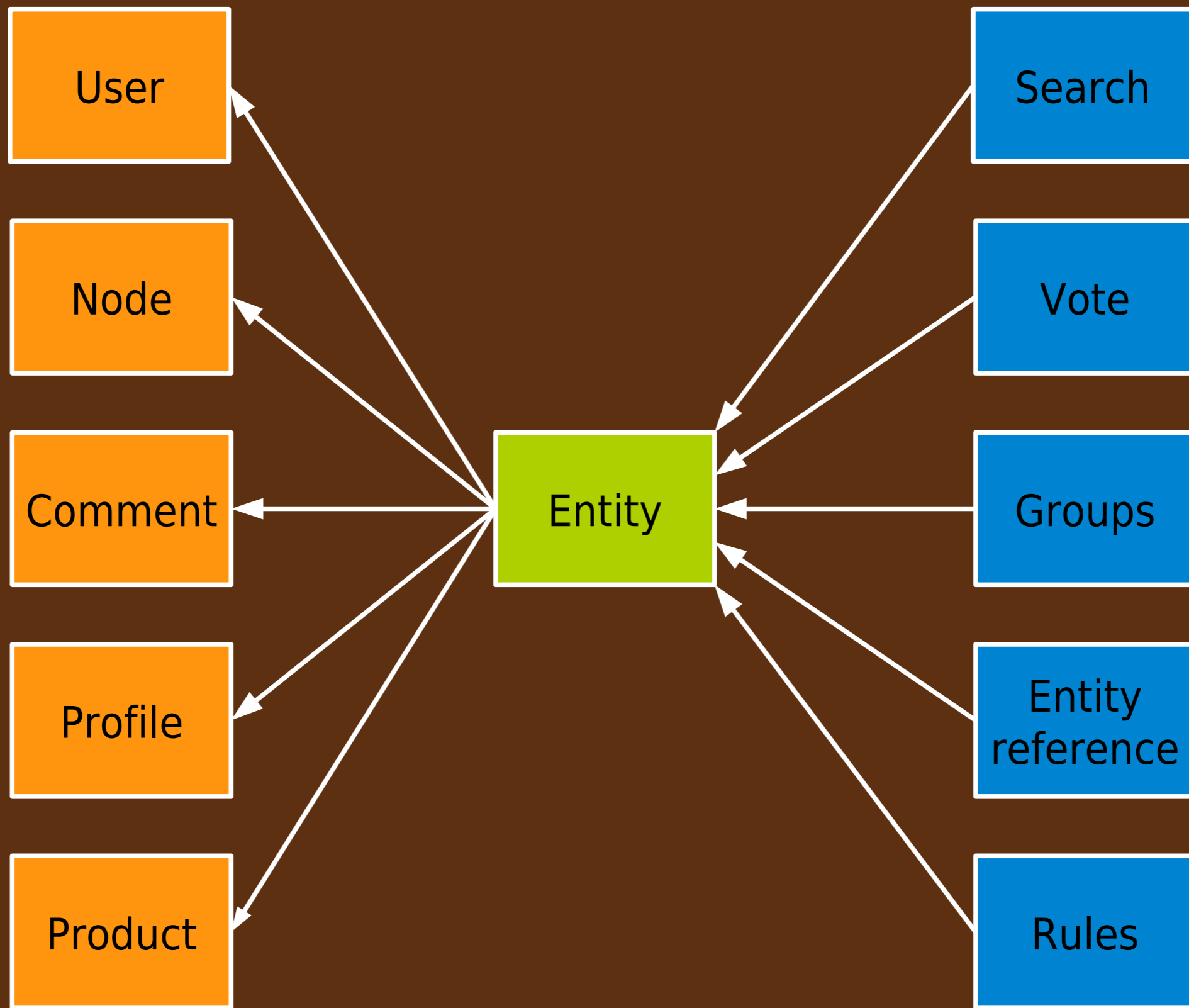
Node modules



Entity modules

Entity API – What for?

Unified way to access data.



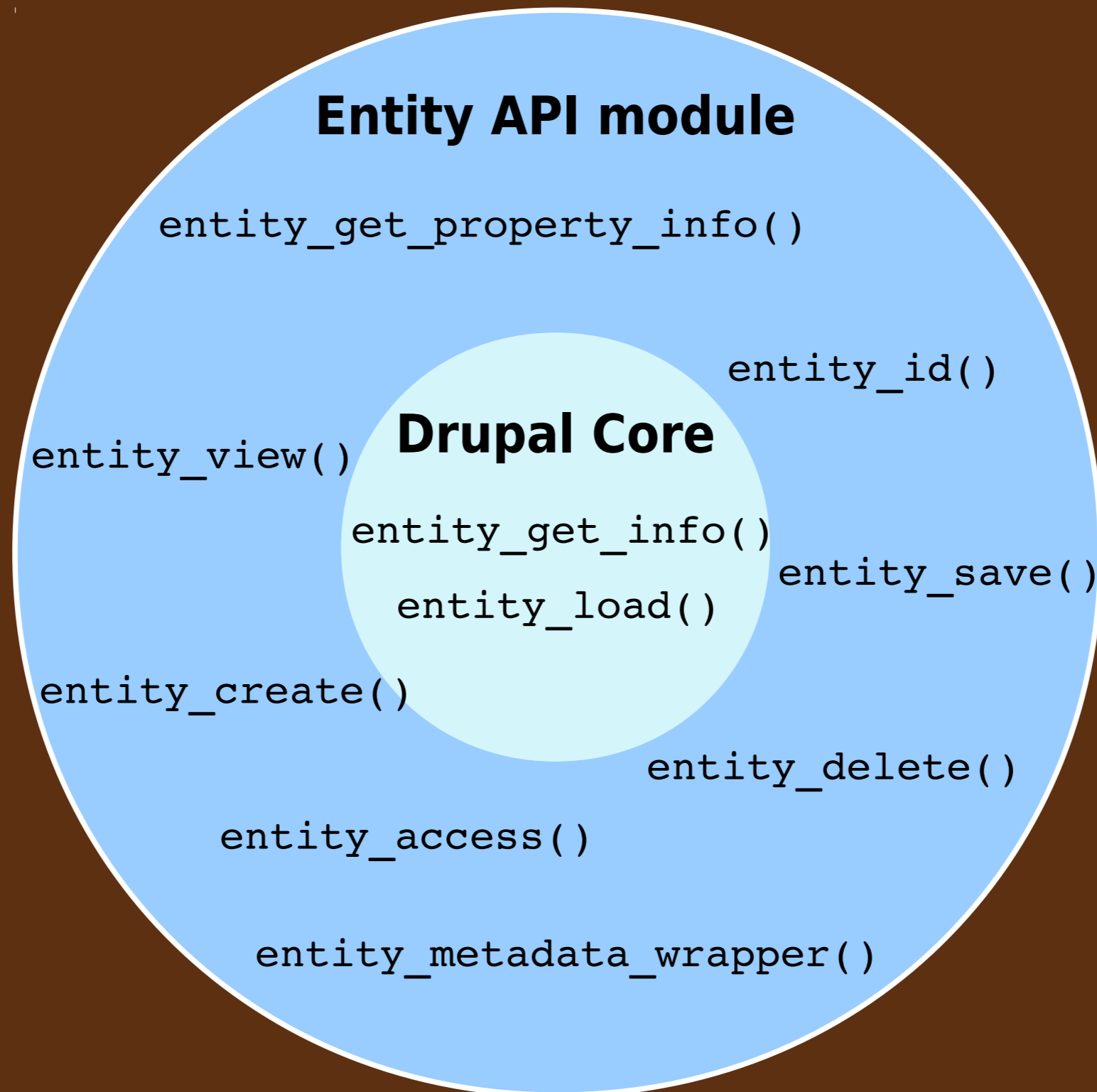
Interacting with entities

Entity API module

Assists you with

- interacting with entities
- and providing new entity types

Entity API Functions



Metadata Wrapper

```
→ $wrapper = entity_metadata_wrapper('node', $nid);  
→ $mail = $wrapper->author->mail->value();  
→ $wrapper->author->mail->set('fago@example.com');  
→ $text = $wrapper->field_text->value();  
→ $wrapper->language('de')->field_text->value();  
→ $terms = $wrapper->field_tags->value();  
→ $wrapper->field_tags[] = $term;  
→ $options = $wrapper->field_tags->optionsList();  
→ $label = $wrapper->field_tags[0]->label();  
→ $access = $wrapper->field_tags->access('edit');
```

Metadata Wrapper

```
$wrapper = entity_metadata_wrapper('node', $nid);  
$mail = $wrapper->author->mail->value();  
$wrapper->author->mail->set('fago@example.com');  
$text = $wrapper->field_text->value();  
$wrapper->language('de')->field_text->value();  
$terms = $wrapper->field_tags->value();  
$wrapper->field_tags[] = $term;  
$options = $wrapper->field_tags->optionsList();  
$label = $wrapper->field_tags[0]->label();  
$access = $wrapper->field_tags->access('edit');
```

Entity property info

```
$properties['mail'] = array(  
    'label' => t("Email"),  
    'type' => 'text',  
    'description' => t("The email address of ..."),  
    'setter callback' => 'entity_property_verbatim_set',  
    'validation callback' => 'valid_email_address',  
    'required' => TRUE,  
    'access callback' => 'user_properties_access',  
    'schema field' => 'mail',  
);
```


Property info?

- Unified access to entity data
- Validation
- Access information

How modules use it

- Drupal Commerce, VBO, OG
- Rules, Search API
- Microdata
- RestWS, WSCClient
- Entity tokens
- Entity Views

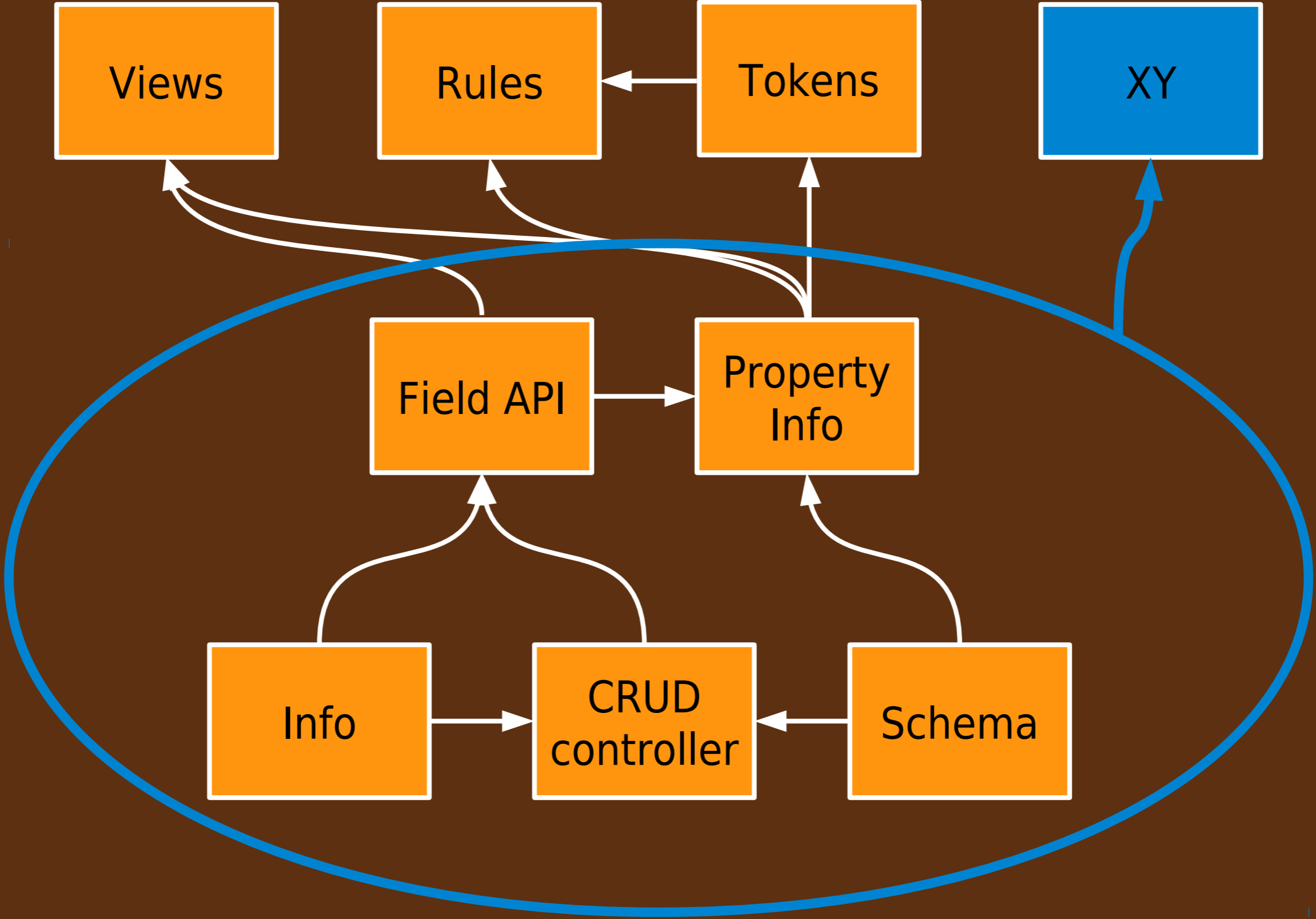
Providing an entity type

- Implement `hook_entity_info()`
- Specify your 'controller class'
- Implement `hook_schema()`

Entity API module

```
$profile = entity_create('profile2', array(
    'type' => 'main',
    'user' => $account,
));
$profile->save();
$profile->delete();
entity_delete_multiple('profile2', array(1, 2, 3));
```

Integrating your entity type

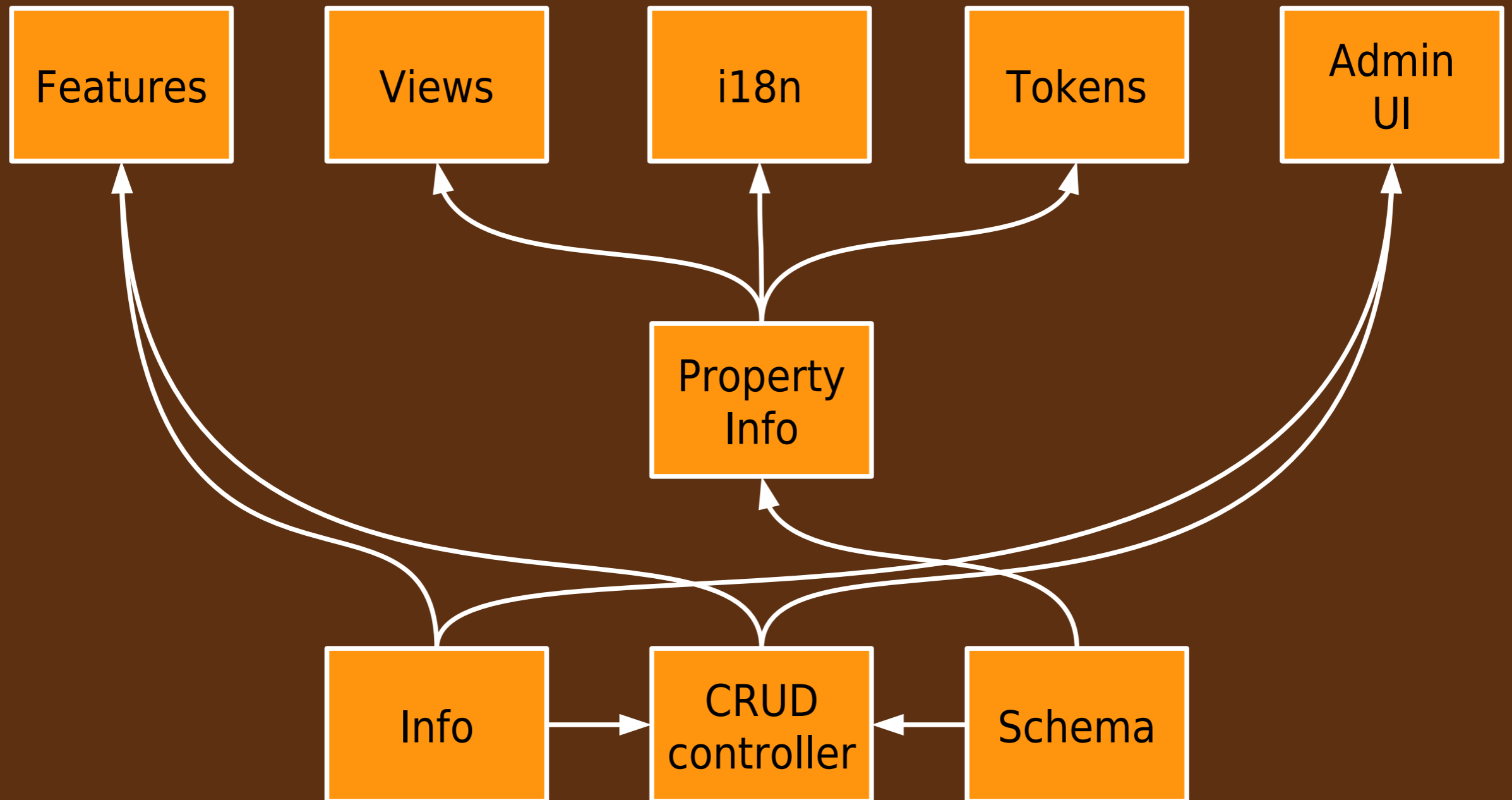


Exportable entities

similar to CTools exportables, but...

- unified CRUD interface
- synced to the DB
- regular CRUD hooks
- Views, Tokens, Features, i18n, ...

Integrating your exportable entity



Apocalypse now?

- Provide an entity
- get a bullshit of not fitting stuff
-
- Find a good metafa, or cite for it.
- is it an elephant or moskito?

Module developers: Think

- what makes it applicable?
- label, URI?
- field?
- custom on,off
- example: no display,no page → no metadata tags

Profile types +

[+ Import profile type](#) [+ Add profile type](#)

LABEL	STATUS	OPERATIONS					
Main profile (Machine name: main)	Custom	edit	manage fields	manage display	clone	delete	export
Community profile (Machine name: community)	Custom	edit	manage fields	manage display	clone	delete	export

Example: Profile2

Profile2:

- Entity type

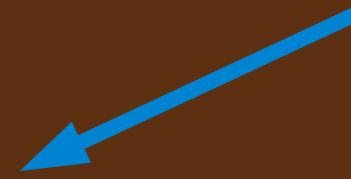
Profile2 types:

- Entity type
- Bundle of Profile2
- Exportable

Profile2 torn apart

- CRUD, Field API
- Permissions, Access
- Profile form, display
- Admin UI, I18n
- Views, CTools, Rules, Tokens, Features, Search API

Your job

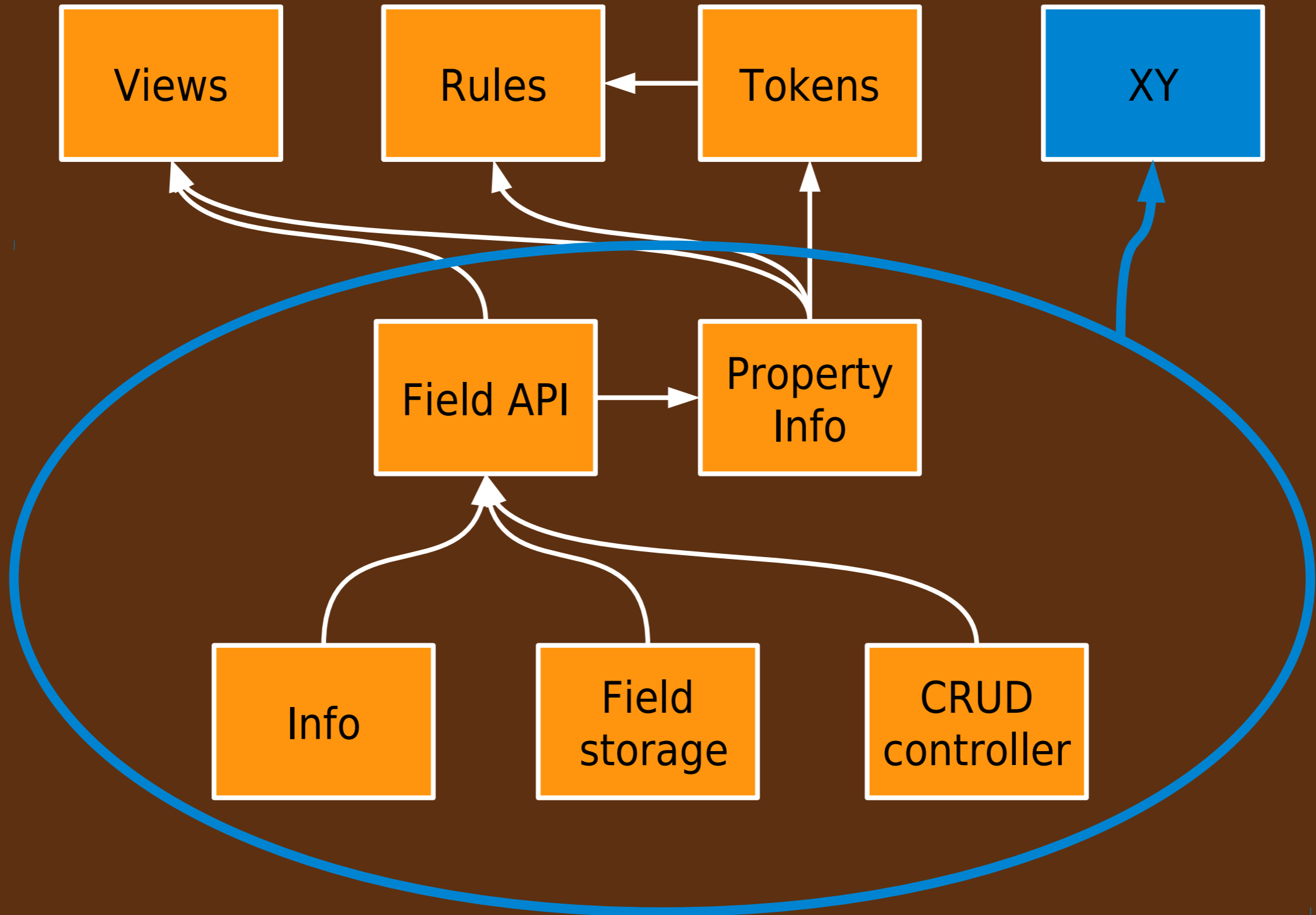


**homework done,
let's enjoy it.**

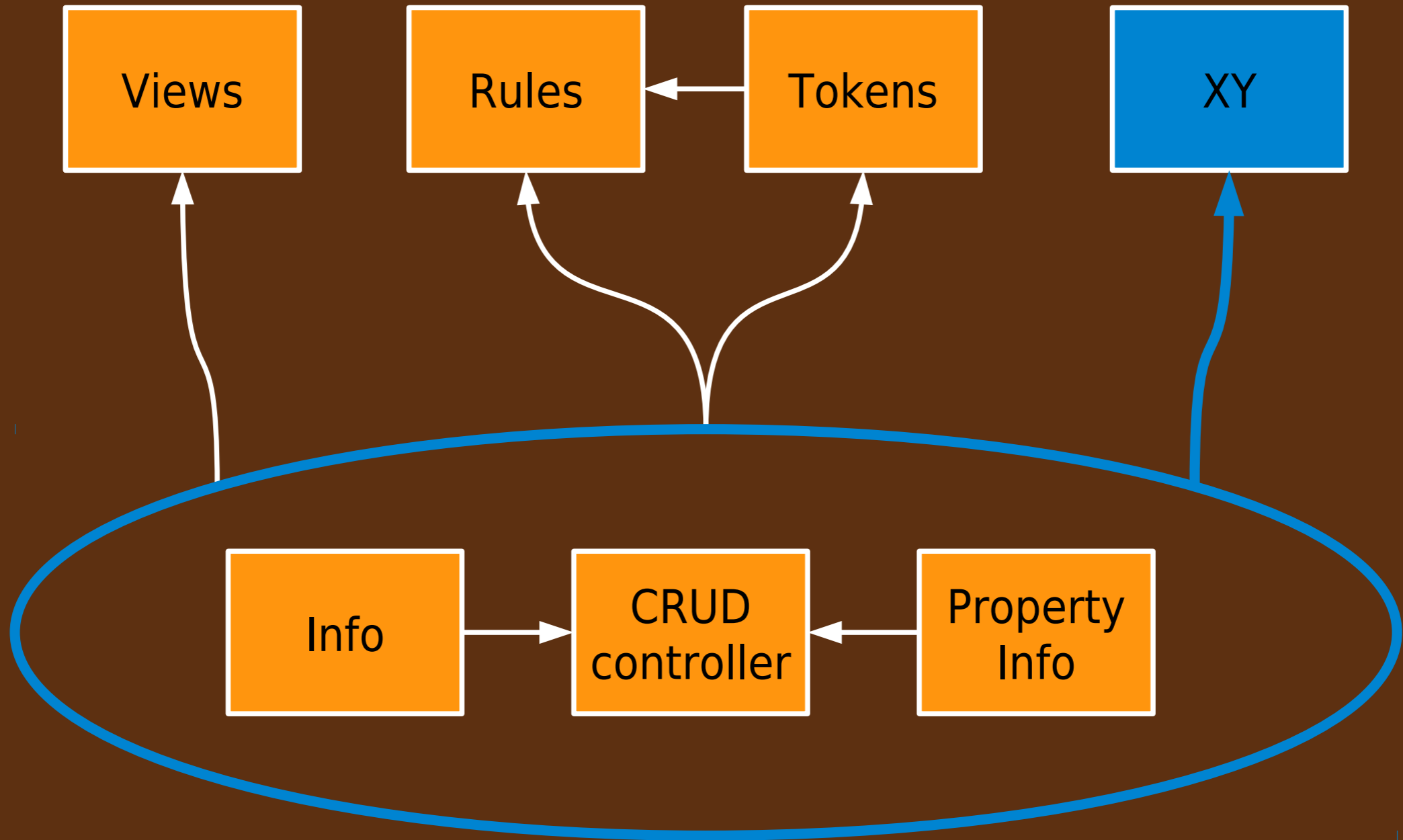
- NoSQL, Doctrine, PHPCR
- Remote entities
- Data integration

**Non-DB entities
+
Fields?**

Non-DB entities



Non-DB entities



What does it buy us?

- Classed CRUD API
- CRUD Hooks
- Tokens
- Entity reference
- Rules, Rules Links
- Search API, Views integration, RestWS
- ...

Drupalize your data!

howto *slide*?

Example...

Google Picasa entities

<http://drupal.org/sandbox/fago/1493180>



Picasa photos

Navigation

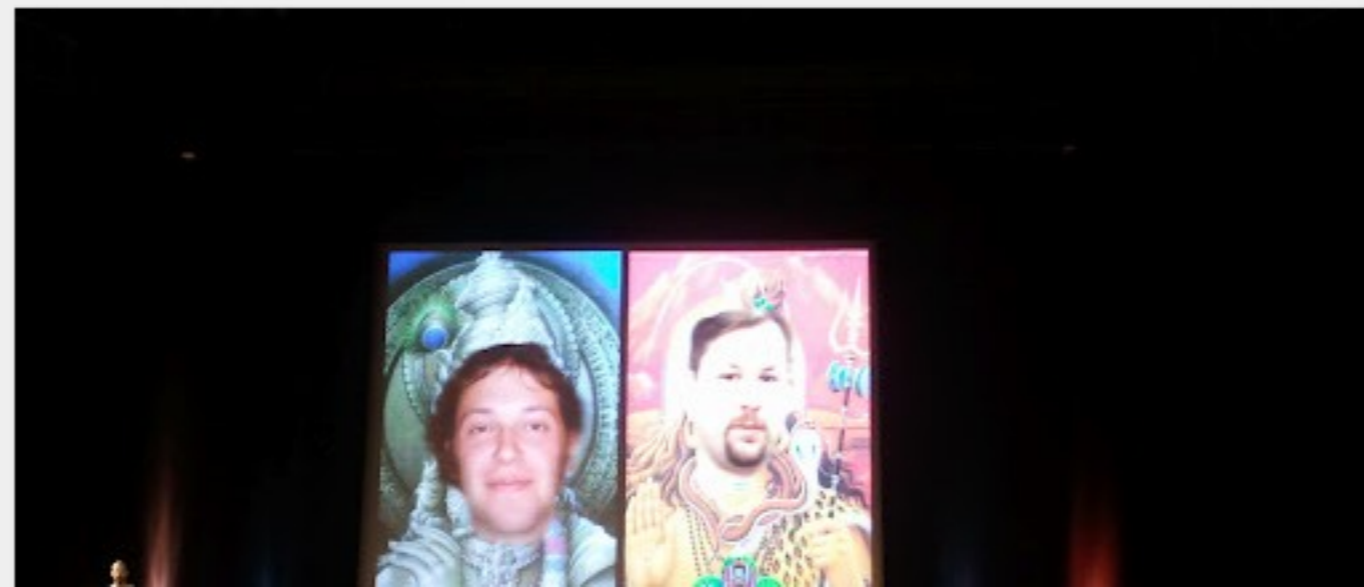
- ▶ [Add content](#)
- [Full picasa photos](#)
- [Picasa photos](#)

Rendered Picasa photo


Album




DrupalconLondon



DrupalconLondon



Navigation



- ▶ [Add content](#)
- [Full picasa photos](#)
- [Picasa photos](#)

Picasa photos

ID	Count of comments	Height	Size	Timestamp	Width	Album	Number of Photos in Album
https://picasaweb.google.com/data/entry/api/user/101657528803489520878/albumid/5720117435286591681/photoid/5720117843868754498	0	1,920	1,532,383	11/13/43617 - 15:20	2,560	DrupalconLondon	13
https://picasaweb.google.com/data/entry/api/user/101657528803489520878/albumid/5720117435286591681/photoid/5720117916990831250	0	1,920	1,650,574	11/17/43617 - 03:30	2,560	DrupalconLondon	13
https://picasaweb.google.com/data/entry/api/user/101657528803489520878/albumid/5720117435286591681/photoid/5720117768079629842	0	1,920	1,588,171	07/24/43615 - 11:13	2,560	DrupalconLondon	13
https://picasaweb.google.com/data/entry/api/user/101657528803489520878/albumid/5720117435286591681/photoid/5720117973672443458	0	1,920	1,521,397	04/19/43618 - 15:40	2,560	DrupalconLondon	13
https://picasaweb.google.com							

Room for improvements

- EFQ Views
- Ready-only mode
- Generated display
- Generated form

Drupal 8

Comments in Drupal 8

```
$comment = entity_create('comment', array(
  'nid' => $node->nid,
));
$comment->save();
echo $comment->id();
$comment->delete();
entity_delete_multiple('comment', array(1, 2, 3));
```

Comments in Drupal 8

```
class CommentStorageController extends
  EntityDatabaseStorageController {
  ...
}
class Comment extends Entity {
  ...
}
class Entity implements EntityInterface {
  ...
}
```

Questions?



What did you think?

Locate this session on the
DrupalCon Denver website

<http://denver2012.drupal.org/program>

Click the "Take the Survey" link.

Thank You!