EZ PUBLISH API

ABOUT ME

- Working at eZ since 2005
- Domain leader for eZ publish
- oms@ez.no
- @dotten

AGENDA

- Some notes about the current status of the API
- Our philosophy & background
- What we are doing
- Community initiative
- Open discussion

eZ®

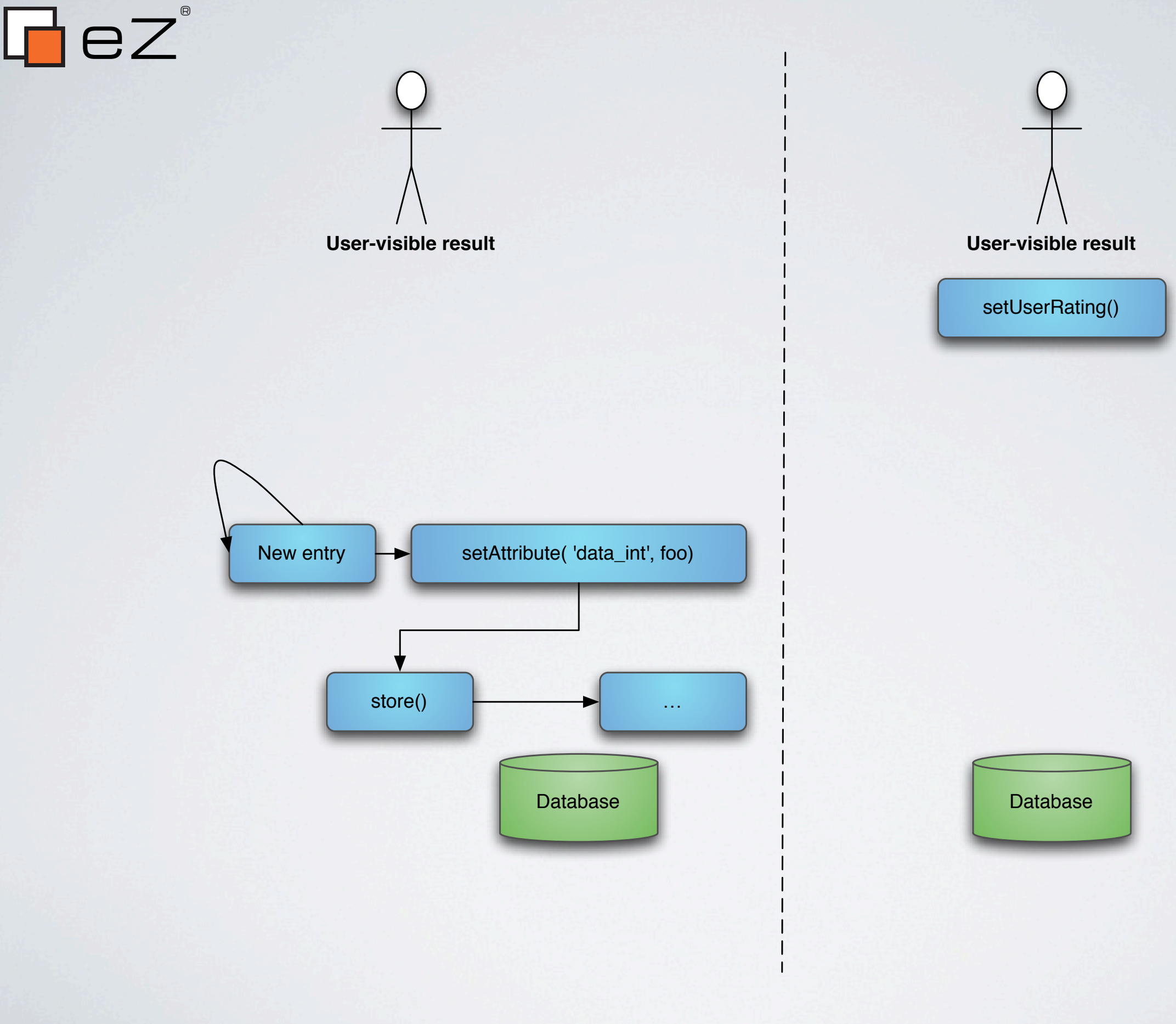
APIs

HOW DOES IT LOOK LIKE

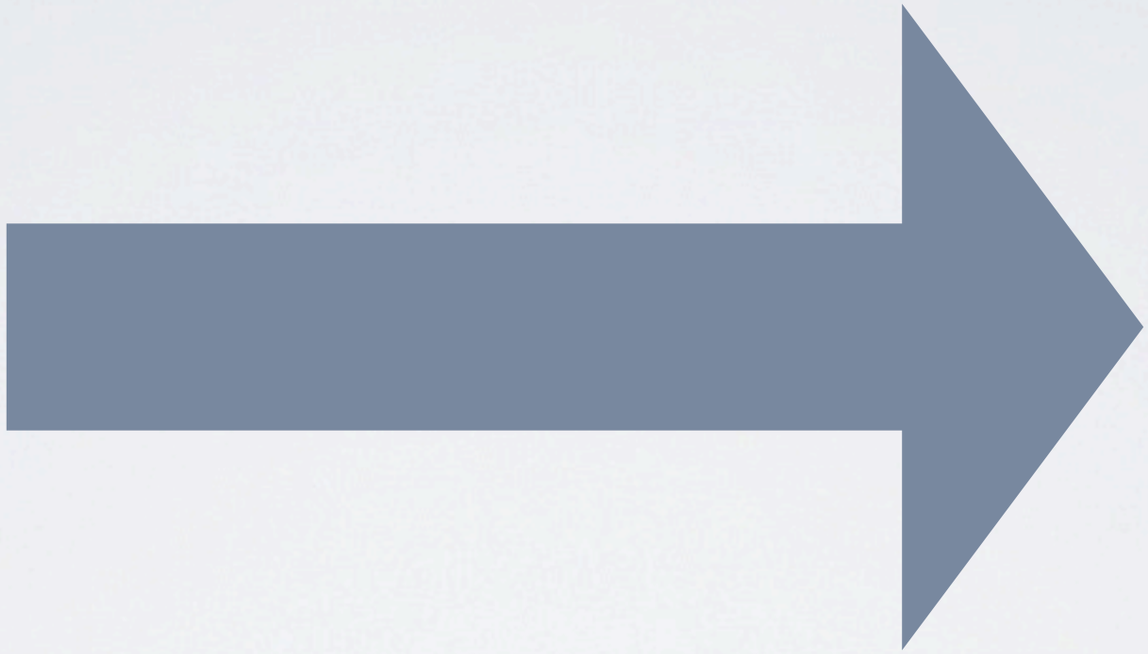
Version	# public methods
4.3	9,190

HOW DOES IT LOOK LIKE

- Good concepts
- Model makes sense
- APIs are somewhat low-level, and influenced by the persistence routines


User-visible result
New entry
setAttribute('data_int', foo)
store()
…
Database
User-visible result
setUserRating()
Database

STATUS



OUR PHILOSOPHY

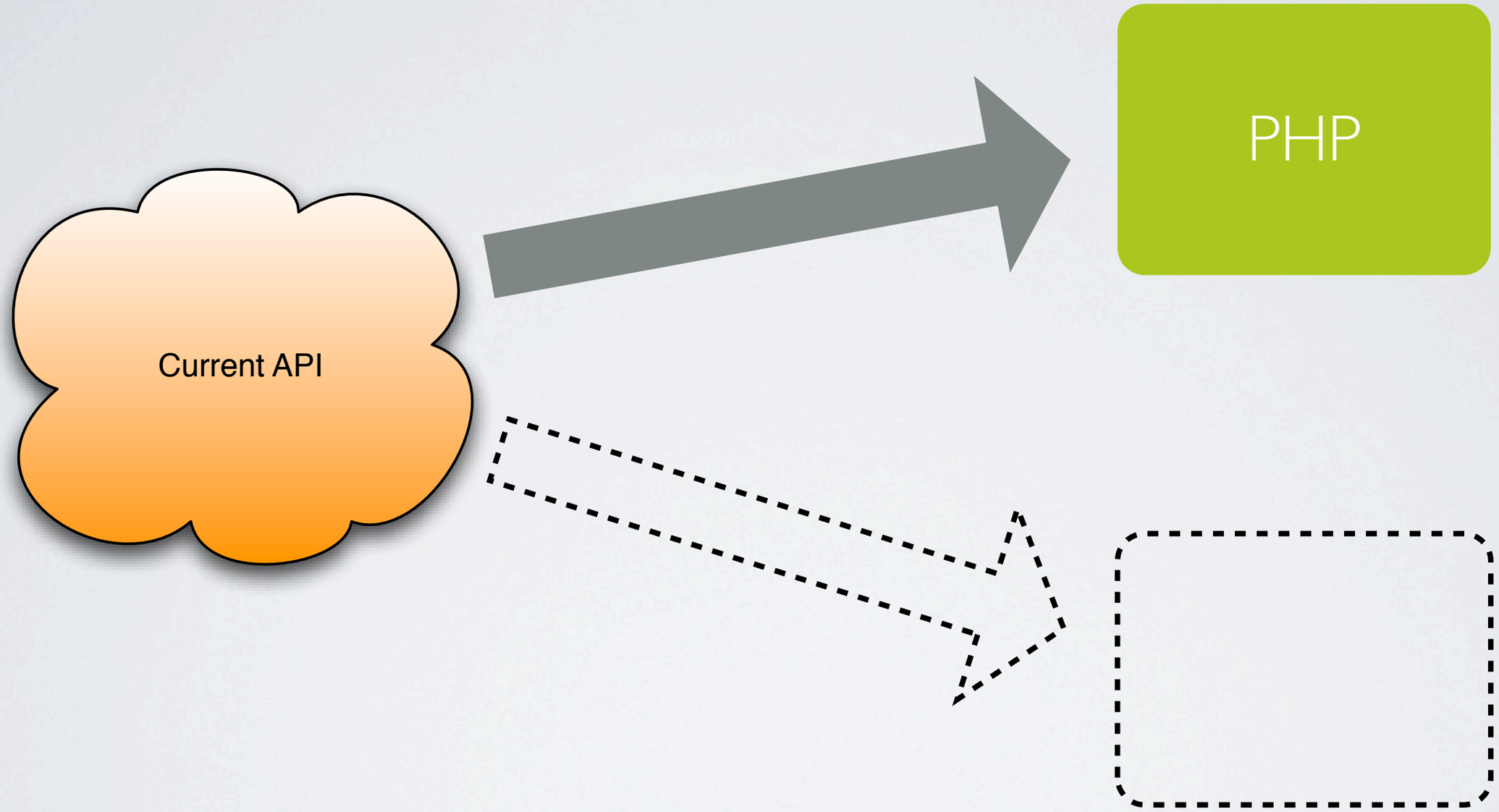
- Consistency and coherence in our model
- Need sound relationships between:
  - Objects ↔ tasks ↔ concepts

WHAT WE WANT

- In general: making simple and common tasks easy
- Stable point of reference
- Still a place for low-level APIs

WHAT WE WANT

- We seek to adopt a DDD-like approach in our new model
- Decouples objects relevant for your model, from the actual storage


PHP
Current API

PHP

- Creating custom functionality, aka extensions
- Adding new behavior and features
- Overriding existing behavior
- Most flexible approach

CHANGES IN THE MAKING

- Decouples logical objects from the storage system
- A common query and retrieval interface
- Search and retrieval can be merged in the API
- Storage-engine neutral

PHP EXAMPLE I

fetching from repository

```php
$c = new ezpContentCriteria();

$c->accept[] = ezpContentCriteria::location()
                ->subtree( ezpContentLocation::fetchByNodeId( 2 ) );

$c->accept[] = ezpContentCriteria::contentClass()->is( 'article' );
```

```php
$articles = ezpContentRepository::query( $c );

foreach( $articles as $article )
{
    echo "English title: {$article->fields['eng-GB']->title}\n";
}
```

PHP EXAMPLE 2

deleting from repository

```
$lister = new ezpContentList();

$lister->searchLocations[] = ezpLocation::fetch( 78 );
$lister->searchLocations[] = ezpLocation::fetch( 81 );

$lister->filters[] = new ezpContentClassIncludeFilter( ‘article’ );

ezpContentService::delete( $lister );
```

PHP EXAMPLE 3

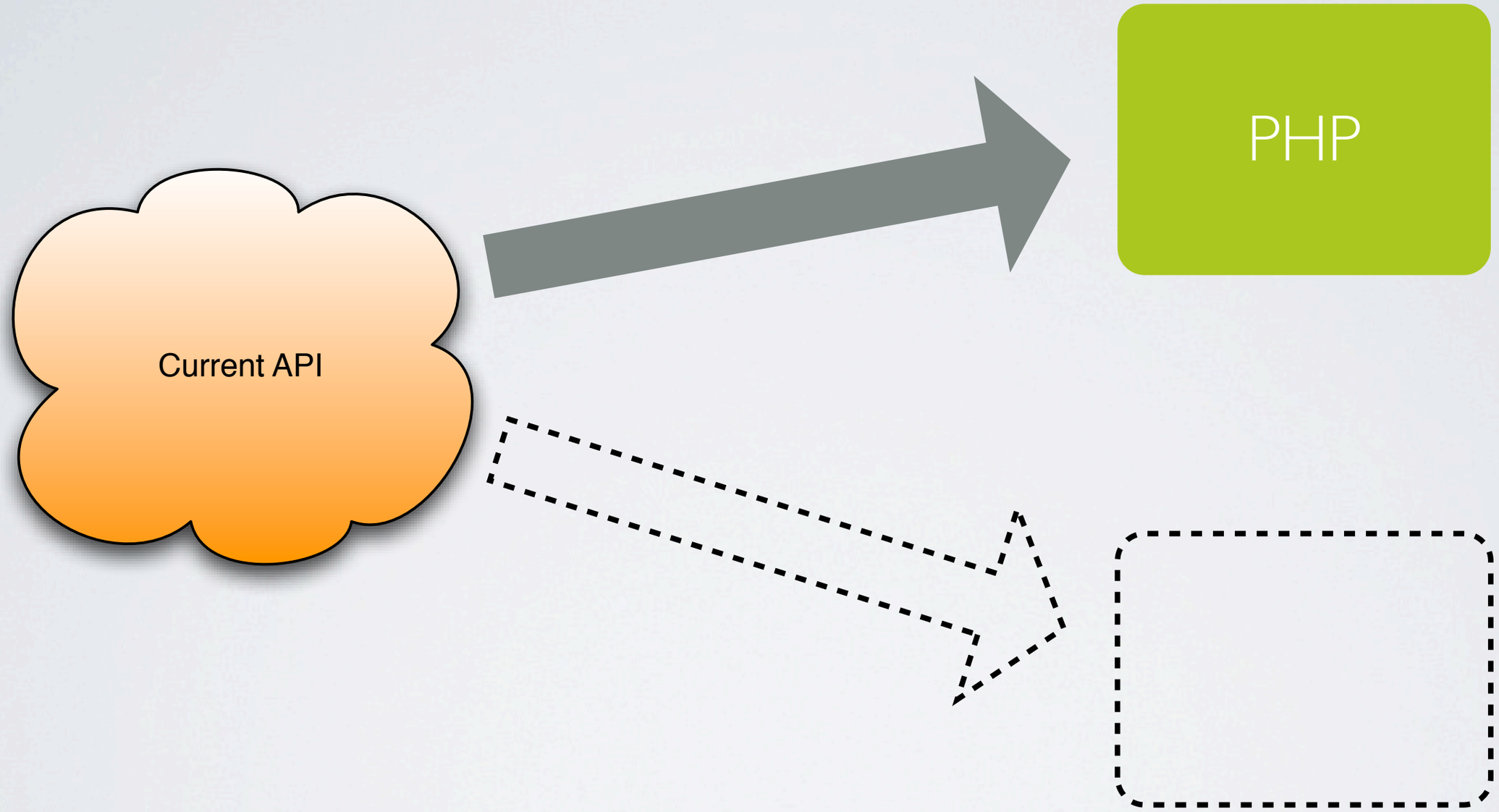
creation

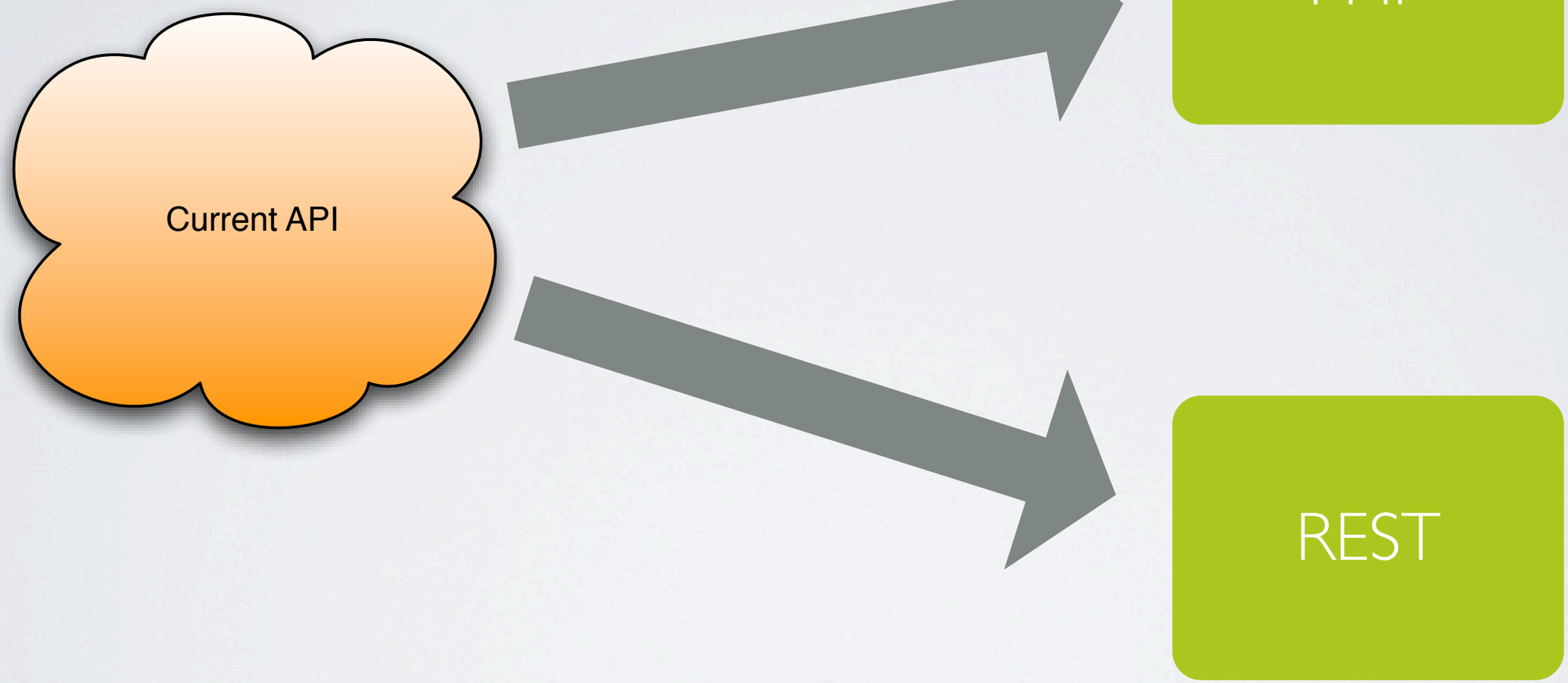
```php
// instantiate a content of type 'article'
$article = ezpContent::create( 'article' );

// add a location for this object
$article->locations[] = ezpContentLocation::fromNodeId( 143 );

// define attributes values
$article->fields->title = 'My article has a title';
$article->fields->body = '<tag>myxmlcontent</tag>';

// At that stage, our article is ready to be published
$publisher = new ezpContentPublisher();
$publisher->publishNew( $article );
```


PHP
Current API


Current API
PHP
REST

REST

- Great for accessing data and services which already exist
- For consuming existing data and leveraging features on external clients

REST

- The REST implementation uses its own end-point
- Pure MVC implementation
- Supports query parameters
- WIP: Unified datatype/field input/output of values

REST WIP

Object mapper

REST WIP

- Speaking HTTP properly
- Important for letting HTTP client libraries handle the request/response
- Conditional GETs
- Playing along nicely with reverse proxies such as Varnish

REST IN THE FUTURE

- Authentication mechanism
- Provide access to extension's APIs
- Expose more features beyond content-consumption

REST EXAMPLE 1

```
GET /api/v1/content/node/<nodeId>/list HTTP/1.1
Host: example.com
```

REST EXAMPLE 2

```
GET /api/v1/content/object|node/<ID> HTTP/1.1
Host: example.com
```

REST EXAMPLE 3

```
GET /api/v1/content/object/<objectId>/field/<fieldIdentifier> HTTP/ 1.1
OR
GET /api/v1/content/node/<nodeId>/field/<fieldIdentifier> HTTP/ 1.1
Host: example.com
```

COMMUNITY INITIATIVE

- We want your input on areas where you've struggled to work with the API
- Any feedback in this area is welcome
- We will put up a feedback form on **share.ez.no** next week
- Contact via other channels as email, twitter is also great

OPEN DISCUSSION