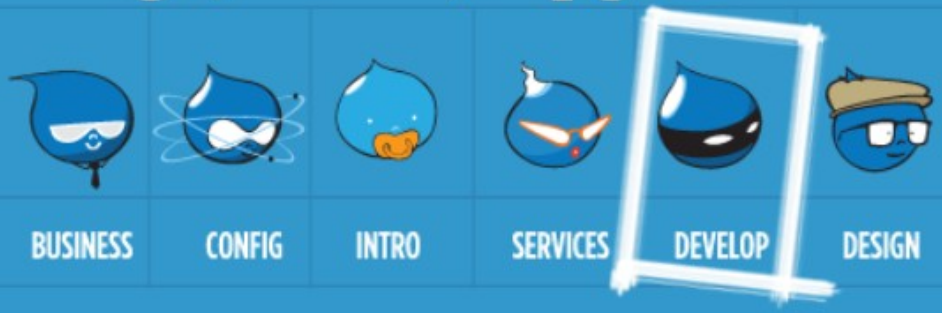
DRUPAL CON CPH

TITLE: The Rules ecosystem

PRESENTER: fago, klausi, sepgil

TRACK:



DATE: 24. aug 10:00

ROOM: Acquia

DRUPALCON COPENHAGEN 2010

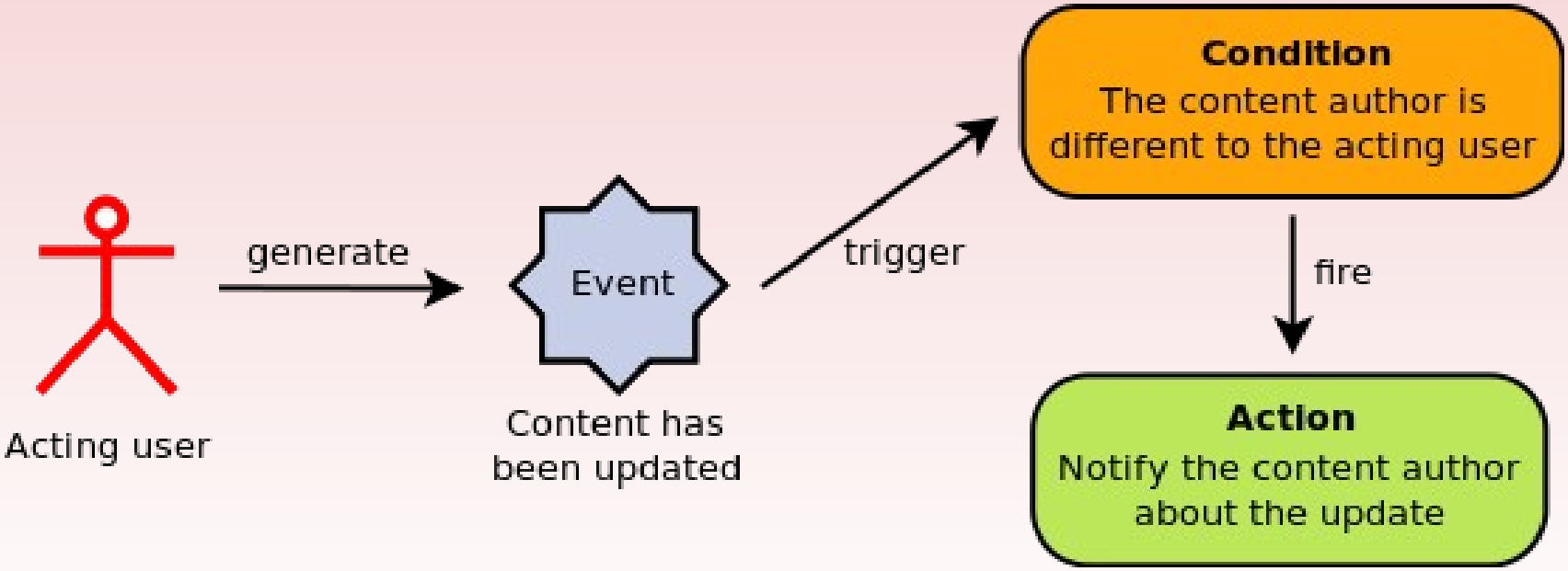
Who?

- Wolfgang Ziegler (fago)
- Klaus Purer (klausi)
- Sebastian Gilits (sepgil)
- epiqo – Austrian based Drupal company
- Drupal Austria user group

epiqo

Rules !?!?

- Reaction rules or so called ECA-Rules
- Event-driven conditionally executed actions



Sessions

- Using Rules 1.x with Drupal 6
  - “Be a happier Drupal developer with Rules” by Johan Falk
  - Thursday, 9:00 Microsoft room
  - focuses on use cases
- This session is about Rules 2.x for Drupal 7
  - focuses more on developers

Outline

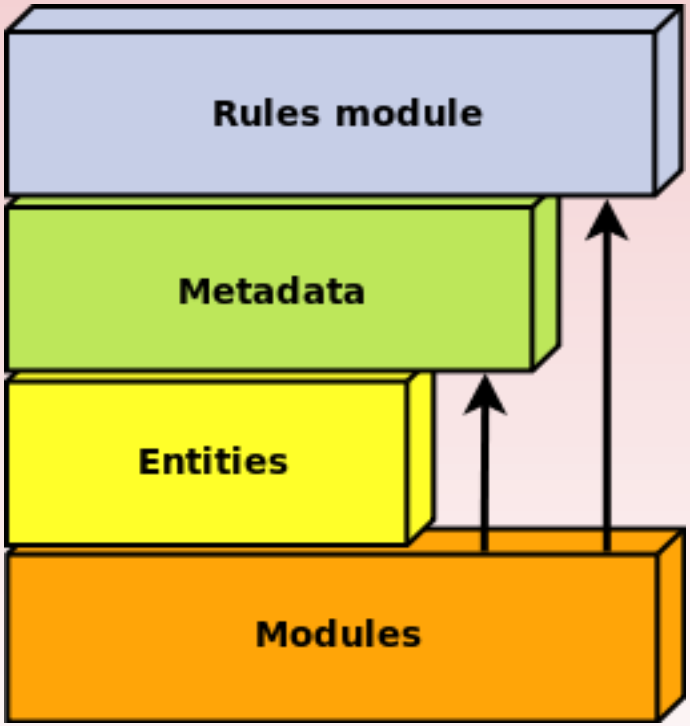
- Rules 2 overview
- New features
- Developing with Rules 2
- Web Service Client (wsclient)
- Rules Web Hooks
- GSoC: Rules Optimization, Rules Transformers

Rules 2 - Overview

- New version for Drupal 7 only
- Re-written to support enhanced use-cases
- Leveraging OOP & Drupal 7 Features
- New, reusable UI with Drag&Drop
- Test coverage

New architecture

Modules may integrate with Rules by providing metadata, entities or directly new events, conditions or actions to Rules.



- Entities

  Entity = node, user, comment, term, ..

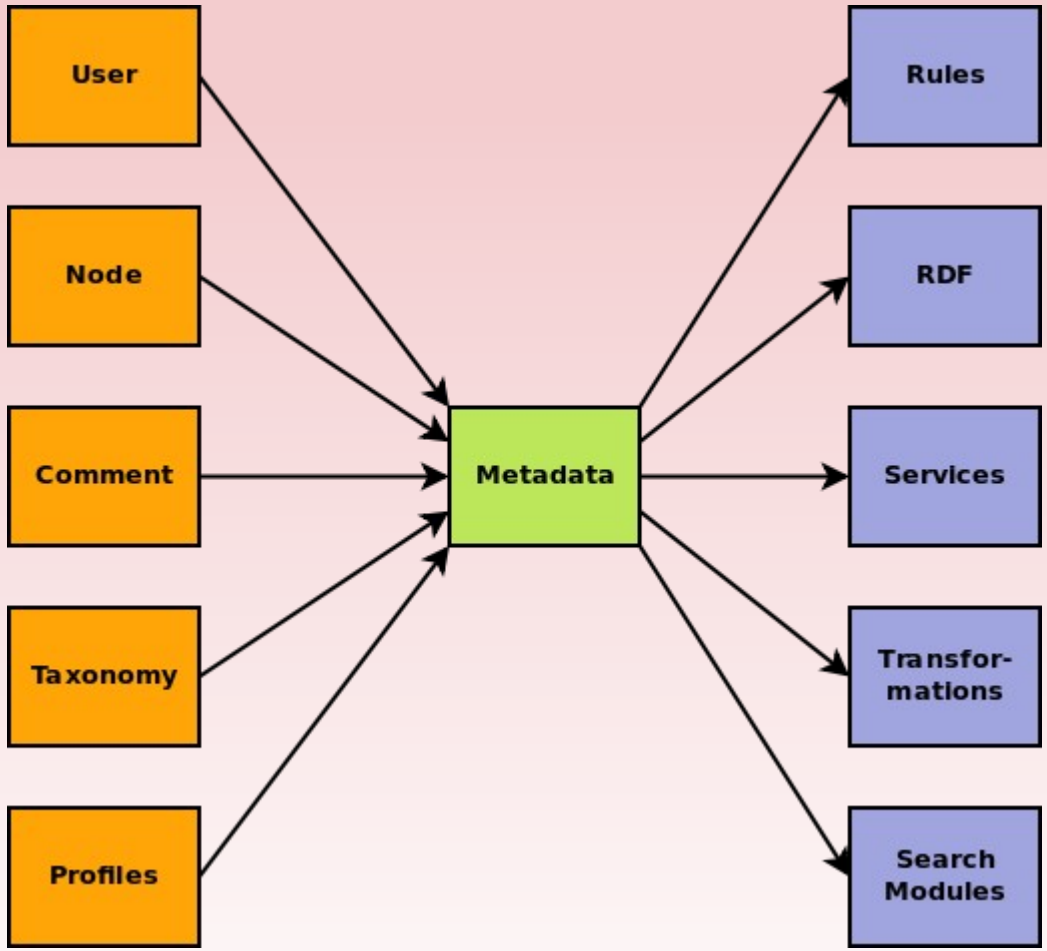
  Rules builds upon Drupal 7's entities for CRUD. Core provides only read, Entity Metadata the rest.

- Metadata

  provides entity properties (e.g. node title, body, ..) and how they can be accessed + CRUD for entities.

CRUD … Create, Read, Update, Delete

Entity Metadata



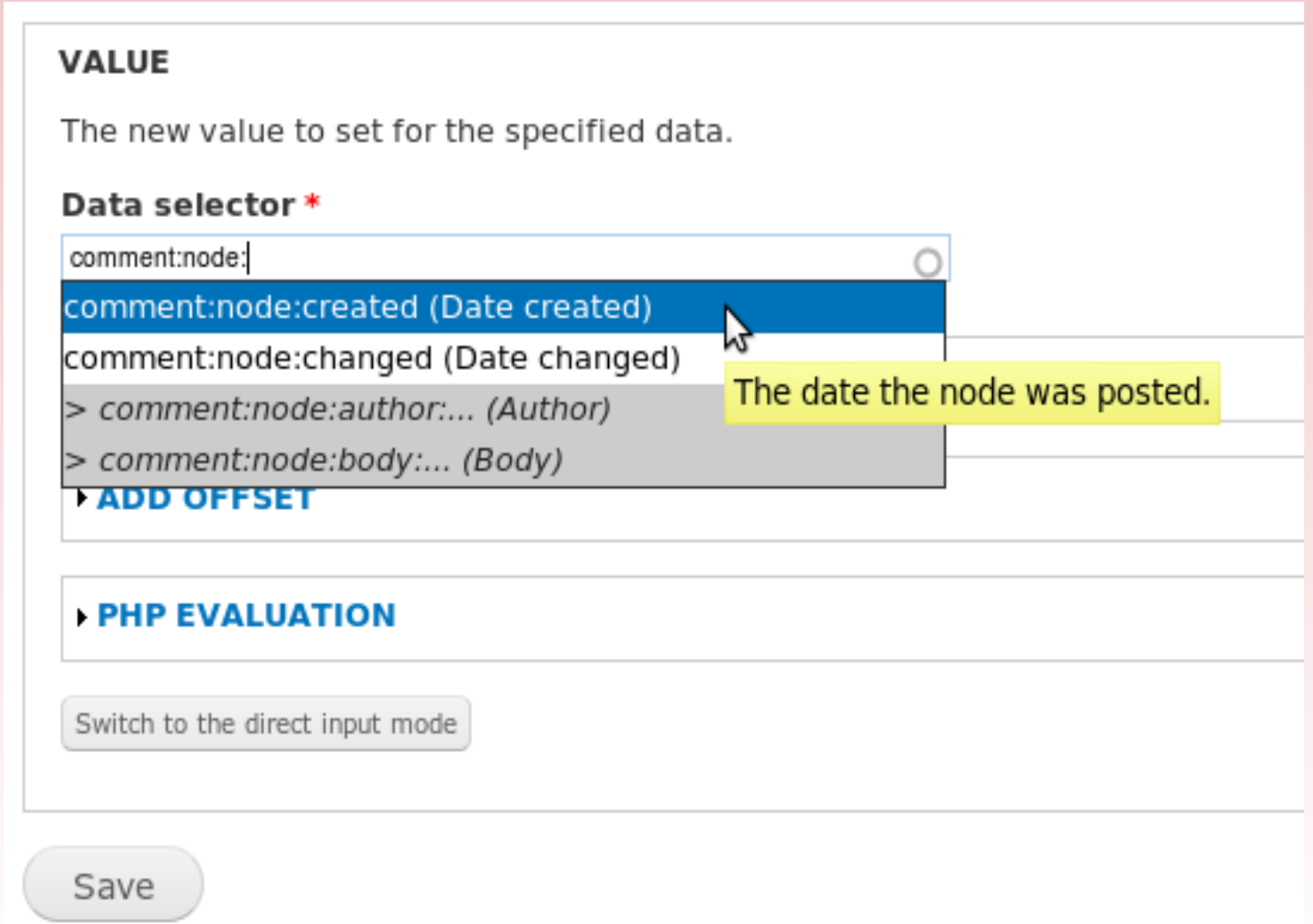
More infos: http://more.zites.net/introducing_entity_metadata

New features

Data selectors 1/2

- 2 argument configuration modes:
  - Direct data input:
    Specify a fixed value, e.g. a number.
  - Data selection:
    Select the data source for the argument value;
    e.g., [node:author:name].
  - Users can switch between both configuration modes per parameter.

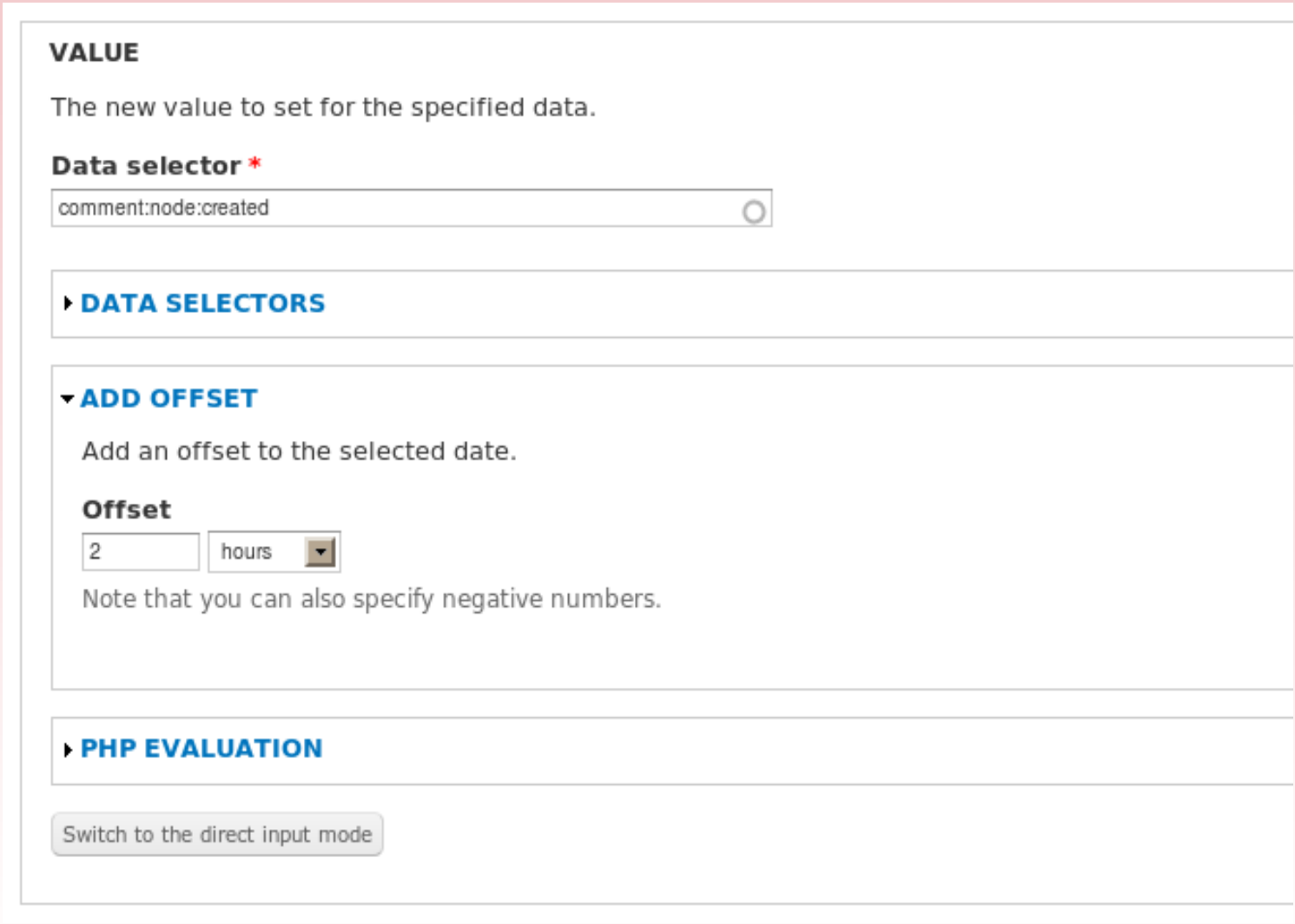
Data selectors 2/2



Data processors (1/2)

- Input evaluation system as known (Supporting token replacements & PHP)
- Input evaluation is only applicable for direct data input, e.g. tokens for textual parameter
- For data selection, data processors are available; e.g.:
  - Move a given date by a week
  - Subtract a number by one

Data processors (2/2)



Data lists

- Rules 2 supports lists of data
- The data type of contained list items may be specified; e.g., list<node>, list<text>.
- Loop plugin:
  - Execute actions on list items
  - Loops over a given list and provides the current item
  - Applies all actions on list each item.

Better integration for Drupal core (1/2)

- Loading actions are much less needed thanks to data properties like `[node:author]`, `[node:field-tags:0]`, `[comment:node]`.
- Full CRUD support for all entity types (nodes, user, taxonomy terms, comments, files, …) based upon Entity Metadata.
- Access to all basic data properties & fields.

Better integration for Drupal core (2/2)

- Set / get each data property with generic actions (data_set, data_is, list_add, list_remove, …)
  → Great for dealing with arbitrary data.
- Specialized conditions/actions for everything that changes site behavior (publish node, block a user, …)

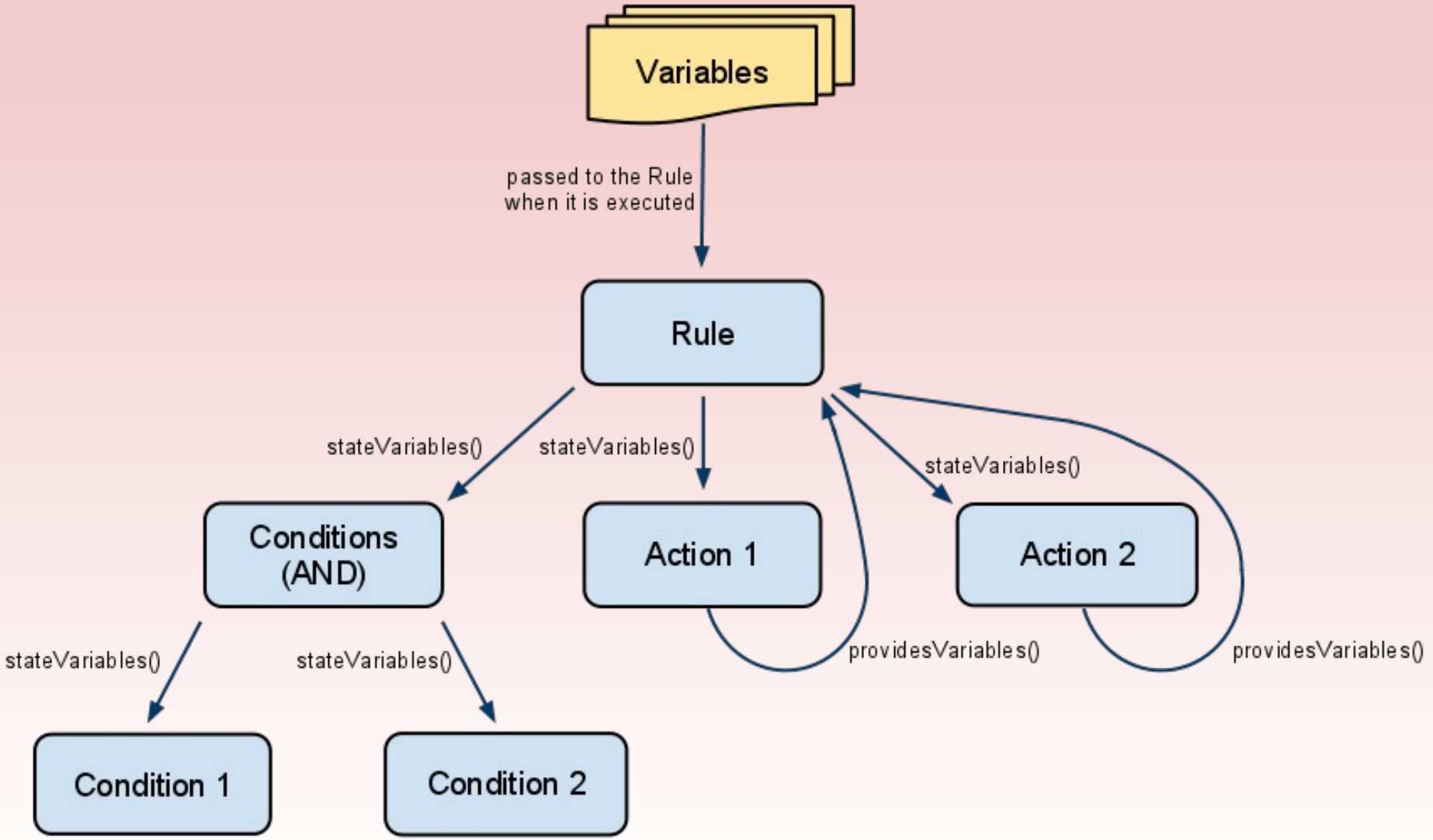
Improved API

- Reusable components (reuse conditions,...)
- Based upon an extensible number of plugins
- A rules configuration is a tree of elements (= configured plugins)
- Each configuration may be saved, re-used and edited on its own.
- Eases programmatic use + writing tests.

Code example

```php
/**
 * Implements hook_default_rules_configuration().
 */
function rules_test_default_rules_configuration() {
  $rule = rules_reaction_rule();
  $rule->label = 'example default rule';
  $rule->active = FALSE;
  $rule->event('node_update')
       ->condition(rules_condition('data_is',
           array('data:select' => 'node:status', 'value' => TRUE))
         ->negate())
       ->condition('data_is', array('data:select' => 'node:type', 'value' => 'page'))
       ->action('drupal_message', array('message' => 'A node has been updated.'));
  $configs['rules_test_default_1'] = $rule;
  return $configs;
}
```

The internals of a configuration



Components

- Configure & reuse components
  - Condition Sets
  - Rule Sets
  - Maybe also pre-configured actions / conditions
- Actions / conditions for the components are provided
- Modules may
  - Predefine components (+ allow altering)
  - Reuse (predefined) components
  - Embed the component configuration UI
  - Embed UI for managing a group of components

Scheduling

- Rules Scheduler module as for Rules 1.x
- Schedule the execution of a component with arbitrary arguments
- Recursion prevention now works across scheduled invocations
- Maybe later: Provide a plugin that allows scheduling actions independent of components

User interface

- Makes use of the tabledrag.js library of core (as known from menu, taxonomy, … admin UI)
- Each plugin may provide its own UI

  → Components come with an reusable UI based upon a defined interface

  → Modules can easily embed parts of the UI

Editing reaction rule "rules_hallo_welt"

Events

EVENT	OPERATIONS
Before saving a comment	delete
✚ Add event	

Conditions

Show row weights

ELEMENTS	OPERATIONS
Content is of type Parameter: *Content:* [comment:node], *Content types:* article	edit delete
✚ Add condition ✚ Add or ✚ Add and	

Actions

Show row weights

ELEMENTS	OPERATIONS
Show a message on the site Parameter: *Message:* hallo [system:current-user...	edit delete
✚ Add action ✚ Add loop	

▸ **SETTINGS**

Save changes

Exporting configurations

- Any configuration may be exported
- New export format based on JSON
  - No PHP evaluation for import required

```
{ "rules_workflow_promote" : {
    "LABEL" : "Promote published content to the front page",
    "PLUGIN" : "reaction rule",
    "REQUIRES" : [ "rules" ],
    "ON" : [ "node_presave" ],
    "IF" : [
      { "entity_has_field" : { "entity" : ["node"], "field" : "field_workflow" } },
      { "data_is" : { "data" : ["node:field-workflow"], "value" : "published" } }
    ],
    "DO" : [ { "node_promote" : { "node" : [ "node" ] } } ]
  }
}
```

Developing with Rules

Integrating with Rules 2

- Provide entity types (+ metadata for CRUD)
- Provide metadata for properties
- Provide conditions, actions and events
  Basically the same as for 1.x, but:
  - Complete parameter description needed
    → Usually no custom configuration forms required
  - Configuration (validation, processing) is independent of form submissions

Providing metadata for properties

```php
/**
 * Implements hook_entity_property_info() on top of node module.
 * @see entity_metadata_entity_property_info()
 */
function entity_metadata_node_entity_property_info() {
  $info = array();
  $properties = &$info['node']['properties'];

  // ...
  $properties['created'] = array(
    'label' => t("Date created"),
    'type' => 'date',
    'description' => t("The date the node was posted."),
    'setter callback' => 'entity_metadata_verbatim_set',
    'setter permission' => 'administer nodes',
    'query callback' => 'entity_metadata_table_query',
  );
  // ...
  return $info;
}
```

Providing action information

```php
/**
 * Implements hook_rules_action_info() on behalf of the user module.
 */
function rules_user_action_info() {
  $items['user_block'] = array(
    'label' => t('Block a user'),
    'base' => 'rules_action_user_block',
    'parameter' => array(
      'account' => array(
        'type' => 'user',
        'label' => t('User'),
        'save' => TRUE,
      ),
    ),
    'group' => t('User'),
    'access callback' => 'rules_user_integration_access',
  );
  return $items;
}
```

Providing action implementation

```php
/**
 * Action: Block a user.
 */
function rules_action_user_block($account) {
  $account->status = 0;
  drupal_session_destroy_uid($account->uid);
}
```

Utilizing Rules 2

- New API eases programmatic use:

```
rules_action('user_block')->execute($account);
```

  → Write test cases!

- Make use of components (condition sets, action sets, rule sets)
- Apply actions on the fly

```
$action = rules_action('system_mail');

// Optional configuration step.
$action->form($form, $form_state);

// Next step: Execute.
$action->execute();
```

Extending Rules 2

- Provide input evaluators
- Provide data processors
- Provide new plugins

→ Extend the rule language

E.g. a different kind of loop, rule set, ..

Developing with Rules 2

- Use Rules for rapid site development
- Rules are
  - easy to understand
  - easy to alter when requirements change
  - well suited for processing by machines → Automate your upgrades!
- Export it to code / create feature modules
- Implementing missing parts by providing new conditions/actions/events

  → Actions form highly reusable components

Web service client

- http://drupal.org/project/wsclient
- Integrates web service operations as Rules actions
- Different endpoints: REST, SOAP, Rules web hooks
- Allows to compose multiple web services in one rule
- Also works standalone as API module to use web services in code
- Examples: Google Translate, Geocoder.us, ...

Web service client description example

```php
/**
 * Implements hook_default_wsclient_service().
 */
function wsclient_soap_test_default_wsclient_service() {
  $service = new WSClientServiceDescription();
  $service->name = 'geocoder';
  $service->label = 'Geocoder.us';
  $service->url = 'http://geocoder.us/dist/eg/clients/GeoCoderPHP.wsdl';
  $service->type = 'soap';
  $operation['label'] = 'Geocode an address';
  $operation['parameter']['address'] = array('type' => 'text');
  $operation['provides']['address_results'] = array(
    'type' => 'list<struct>',
    'property info' => array(
      'zip' => array('type' => 'integer'),
      'street' => array('type' => 'text'),
      'city' => array('type' => 'text'),),);
  $service->operations['geocode_address'] = $operation;
  $services[$service->name] = $service;
  return $services;
}
```

Web service client call example

```
$service = wsclient_service_load('geocoder');
$result = $service->geocode_address(
  '1600 Pennsylvania Av, Washington, DC');
$zipcode = $result[0]->zip;
// $zipcode is now '20502'
```

Web service client TODO

- UI for input of service descriptions
- Automatic parsing of WSDL files
- Exportable service descriptions for sharing
- Web service descriptions are exportable/fieldable entities themselves
- Polish the API and restructure the Rules integration

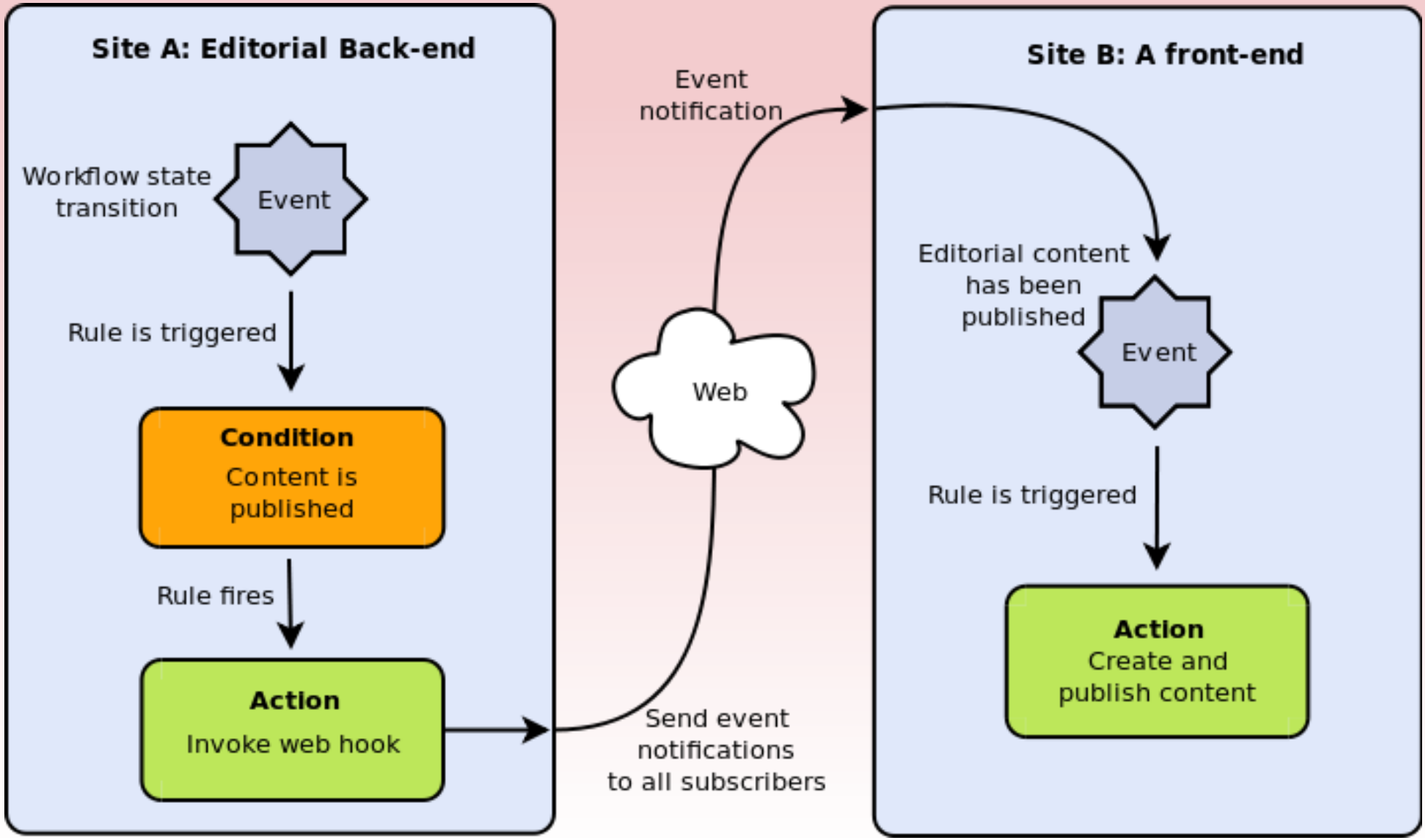
Rules web hooks

- http://drupal.org/project/rules_web_hooks
- Web hooks:
  - Sites may subscribe to web hooks and get notified via a POST request when the hook is invoked
- Allows sites to react on remote events
- Rule-based interaction of different Drupal sites
- Push instead of poll → real-time communication
- Example: Invoke a web hook when a term is added on site A → React + add the term on site B

Rules web hooks internals

- Remote sites subscribe to a web hook
- Subscription mechanism of wsclient to get notified about events
- Relies on services.module to expose entities and metadata
- A web hook specifies parameters that are pushed to subscribers
- TODO: UI

Rules web hooks distributed workflow



GSoC: Rules optimization

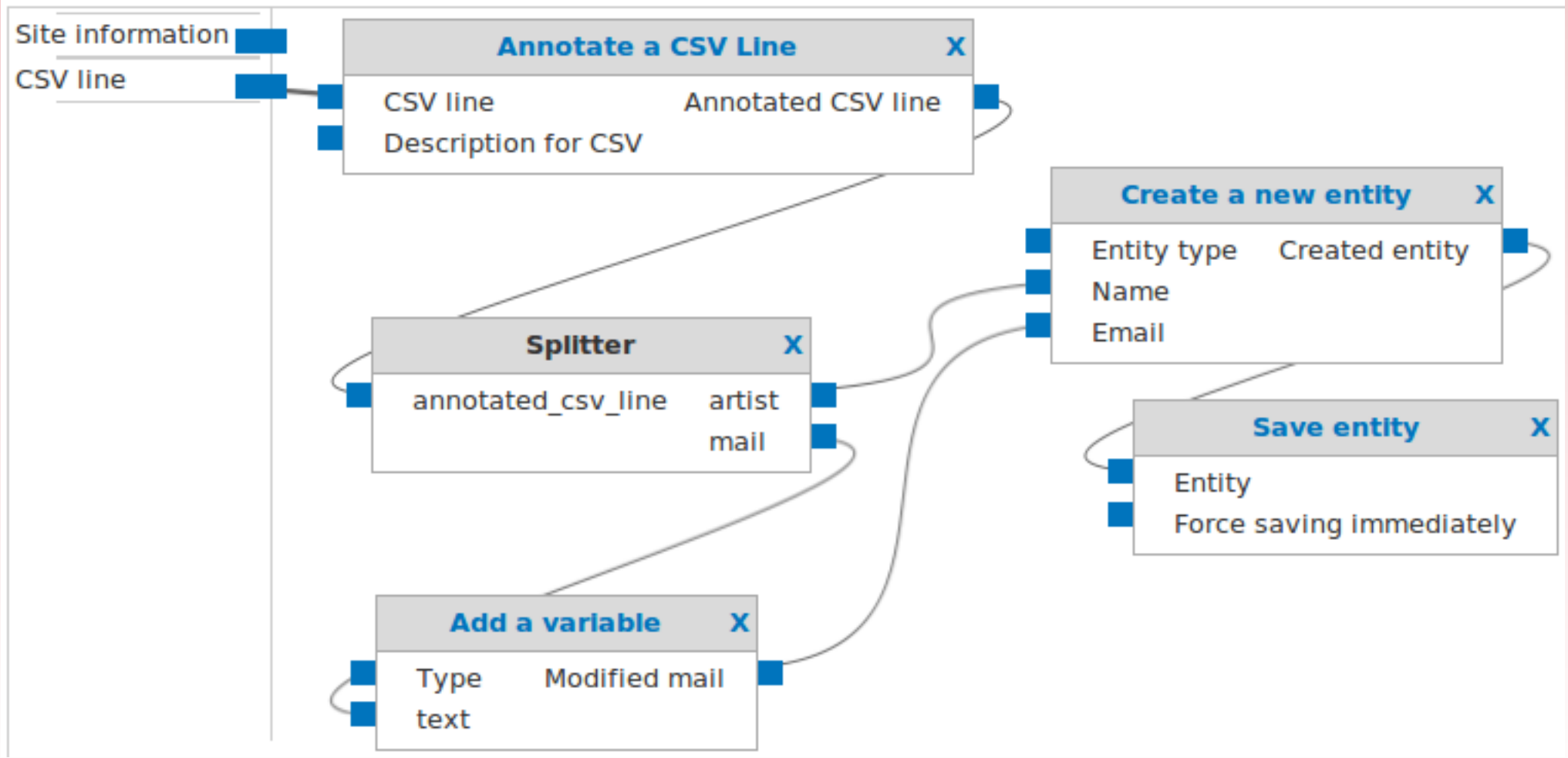
- http://drupal.org/project/rules_optimization
- Builds a decision tree to improve condition evaluation
- Applies to rule sets and event sets
- Conditions are only evaluated once

- Further enhancements, e.g. if the node type is "story", it cannot be "page" in the next condition
- Benchmark test shows considerable improvement on large rule sets (500 rules)

GSOC Rules Transformers

- Goals
  - Map data between structures, schemes
  - Yahoo Pipes like Javascript UI
  - Import, Export
- Rules Component
  - Use it in other Rules
  - Usable by the Rules API
  - Other modules

GSOC Rules Transformers



GSOC Rules Transformers

- Current State:
- Transformer Pipelines
  - XML: Import & Export + XPath queries
  - CSV: Import & Export
- Transformers UI
  - Mapping works
  - Usability flaws
  - Slow Javascript, ajax callbacks & site reloads

GSOC Rules Transformers

- Further use cases
  - Map required data structures for wsclient actions
  - Integrate with data migration tools (Migrate)
- Future
  - Support more data structures
  - Improve the usability
  - Faster UI

Thank you!

BE AWESOME!
EVALUATE THIS SESSION

http://cph2010.drupal.org/node/11413

DRUPALCON
COPENHAGEN