

Web Services

- A Web service is a software system designed to support interoperable machine-to-machine **interaction over a network** (W3C definition).
- Web service are self-contained, loosely coupled and have a platform independent interface.
- Web service types:
 - **SOAP** oriented services (WS* services)
 - **RESTful** web services
 - **REST-RPC** hybrids
- Modern web applications need integration with other **remote software components** and therefore facilitate web services.

Composition

- Web service composition is the process of arranging multiple web services in one **workflow**.
- Complex applications can be built from combining smaller web service components.
- Types of composition:
 - Orchestration (e.g. WS-BPEL)
 - Choreography (e.g. WS-CDL)

Drupal



- is used for building web sites and web applications
- **Web Content Management System**
- Web framework
- PHP, procedural and object-oriented
- Free & Open Source Software (GPL)
- Extensible, modular, configurable
- Powers 1% of all web sites

Objectives



- Integrate a common **web service abstraction layer** into Drupal.
- Map web service invocations to configurable Rules actions.
- Accomplish web service composition with the Rules module as basis. Take the data flow between services into account and compare the Rules language features to traditional composition languages like WS-BPEL.
- Provide a **user interface** to administer and invoke web services without programming effort.
- Automatically parse WSDL metadata for SOAP services.
- Make web service descriptions exportable for sharing with other Drupal sites.
- Implement an **automatic translation** use case that comprises a workflow with the invocation of several web services.

Web Service Client



- Is a module for Drupal (short name: wsclient).

- **Web service descriptions** specify operations, data types and settings for a service.

- Web service descriptions are stored as Drupal **entities**. They can be created in code or entered via the user interface.

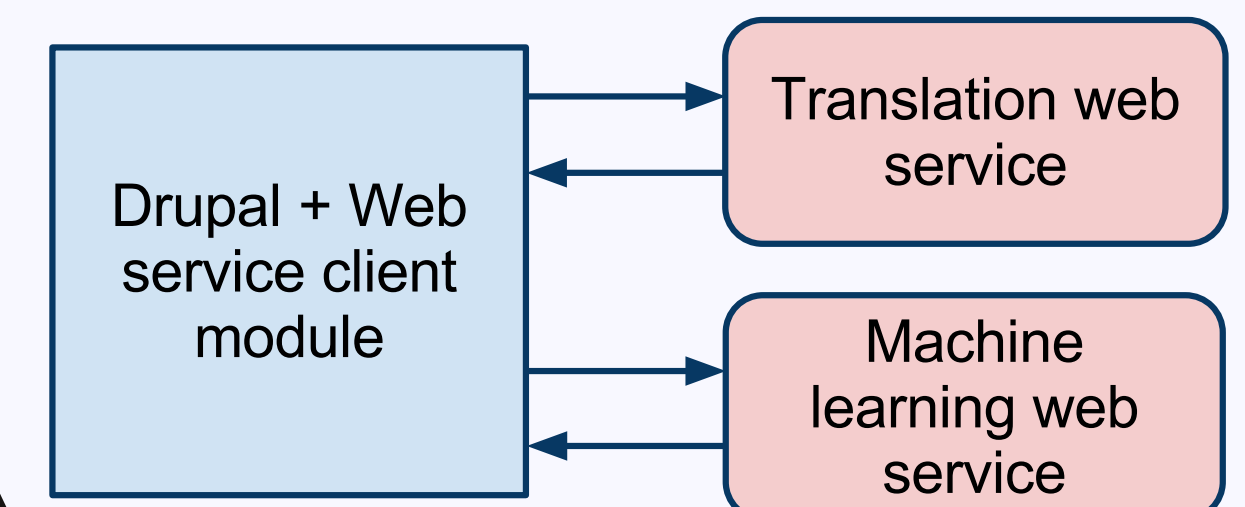
- Different web service types (REST, SOAP) are modeled as **endpoints** that are independent from the web service description.

- Web service composition is realized with **invocation actions**, data structure creation actions and the data selector in Rules.

- Arbitrary web service calls to different endpoint types and a **deep integration** with Drupal's internal data is possible in a workflow.

- Web service descriptions for SOAP services can be automatically extracted from WSDL files.

- The **export** of a web service description is represented in **JSON** and can be imported elsewhere.



Rules



- The Rules module implements a **workflow system** for Drupal.

- Events can trigger a rule where conditions are evaluated and upon success actions are executed (**Event-Condition-Action rules**)

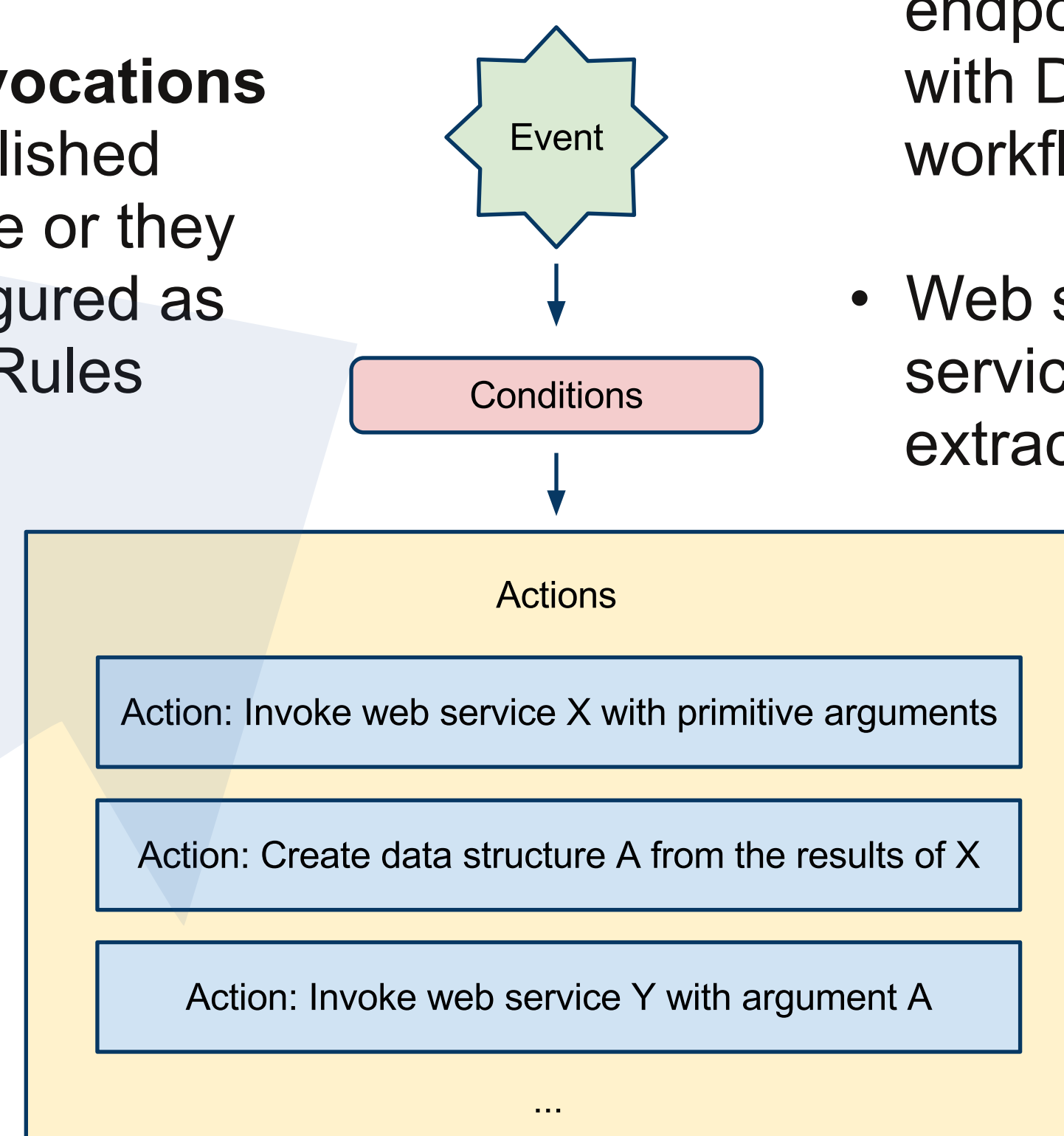
- Rules relies on the **metadata** of complex data structures provided by the Entity module for processing data.

- Intelligent data selectors and a comprehensive user interface assist with the creation of rules.

- **No programming expertise** is required in order to configure and manage workflows with Rules.

- Third party modules can provide their own events, actions and conditions to extend the capabilities of Rules.

- **Web service invocations** can be accomplished directly in code or they can be configured as action in a Rules workflow.



<http://drupal.org/project/wsclient>

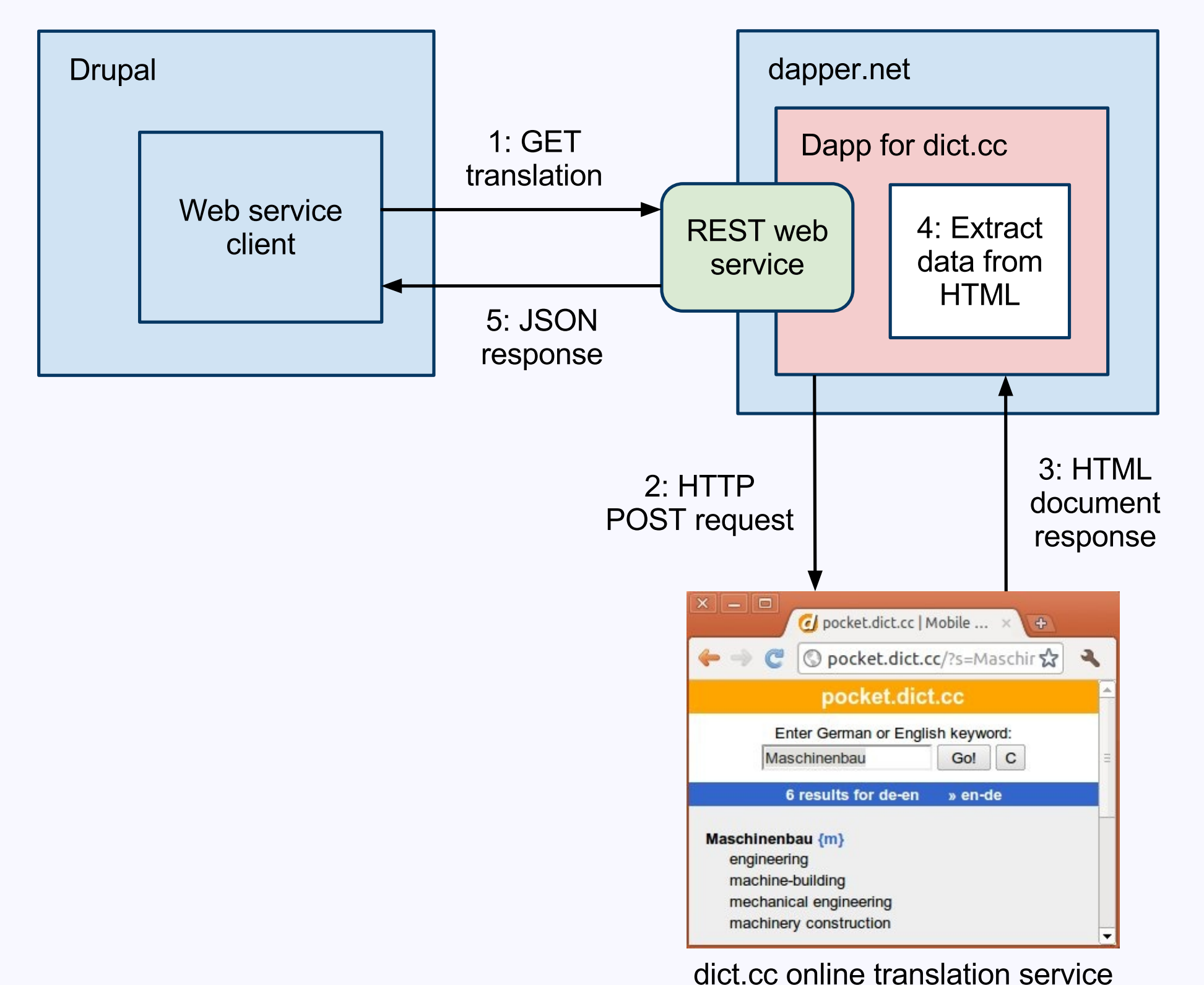
Use Case

- A workflow that communicates with **multiple web services**.

- English translations are retrieved for German taxonomy terms.

- The translations are ranked by a **machine learning** web service.

- Common translation services are accessed with the help of a **web data extraction** service (dapper.net).



Conclusion

- Abstracting different web service types is possible, although the mechanism must be highly extensible.
- The scientific topic of web service composition was successfully applied to a real world scenario.
- Web service integration in Drupal was accomplished by configuration instead of programming effort.
- Developments for generic Drupal entities are a benefit for other modules as well (e.g. the administration UI).

