MT Server Land

DFKI LT's open-source MT network architecture

Christian Federmann, Andreas Eisele

Overview

- Python-based network architecture for MT
- central "*broker server*" dispatches requests
- distributed "*worker servers*" handle MT tasks
- Browser-based access for end users
- API access for integration into custom apps
- Open-source project hosted at GitHub

Our Vision

Motivation

- Make MT from ongoing research accessible to everyone
- Build up a shared MT infrastructure for our projects at DFKI's LT lab
- Allow easy translation using multiple MT engines and/or configurations
- Connect to external applications

Core Requirements

- Single entry point to multiple MT engines for multiple users
- Many language pairs, multiple engines per pair
- Simple web-based access and APIs

Important Features

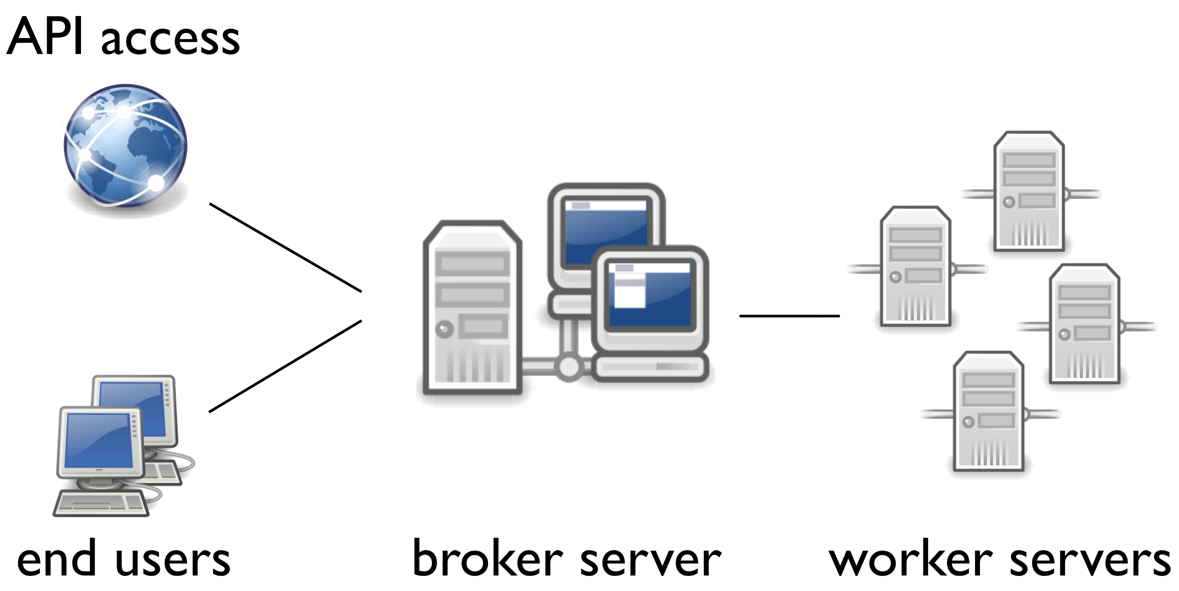
- Scalability via distributed implementation
- Robustness wrt. failures in all modules
- Keep administrative effort low
- Management of user roles and privileges

Advanced Functionality

- Give access to intermediate results
- Allow fine-grained influence on behaviour of MT engines
- Make auxiliary processing steps (segmentation, normalisation) accessible via uniform interface
- Support needs of interactive translation, incremental training, and other hot topics of ongoing research

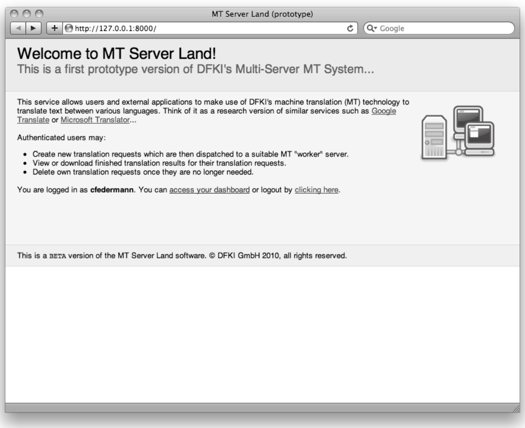
System Architecture

System Architecture



End User Access

- Browser-based interface
- Password protected
- Allows to create new, view finished or delete translation requests
- Results downloadable
- Implemented in django
- Hosted using lighttpd



Welcome to MT Server Land!
This is a first prototype version of DFKI's Multi-Server MT System...

This service allows users and external applications to make use of DFKI's machine translation (MT) technology to translate text between various languages. Think of it as a research version of similar services such as Google Translate or Microsoft Translator...

Authenticated users may:

- Create new translation requests which are then dispatched to a suitable MT "worker" server.
- View or download finished translation results for their translation requests.
- Delete own translation requests once they are no longer needed.

You are logged in as **cfedermann**. You can access your dashboard or logout by clicking here.

This is a BETA version of the MT Server Land software. © DFKI GmbH 2010, all rights reserved.

API Access

- Token-based authentication for security
- Uses HTTP connections (GET, POST, DELETE)
- Several export formats (JSON, YAML, XML)
- Can be used with non-Python frameworks
- It is possible to throttle access to functions
- Uses Google protocol buffer serialization

API Access, cont'd

- API methods either available directly from the django application via HTTP calls
- Or via an XML-RPC service wrapper
- We also plan to extend the export formats to include protocol buffer messages (as these are used anyway by the application)
- Implemented in `dashboard/api`

Object Models

Object Models

- Defined in `dashboard/models.py` and `dashboard/api/models.py`
- 2 central models:
  - `WorkerServer`, models an external worker server that exports MT functionality via XML-RPC
  - `TranslationRequest`, models a translation request, including related information

Worker Servers

- `WorkerServer` implementation includes information on supported language pairs and status methods (`is_alive`, `is_busy`...)
- Translation requests are serialized into a Google protocol buffer "message" which allows for easy serialization of data
- Our `.proto` definition contains request id, source/target language, source/target text and additional "packet data"

Translation Requests

- `TranslationRequest` implementation allows to create a translation "job" on a suitable worker server
- We first generate an "underspecified" protocol buffer and send the serialized data to the worker server
- All communication relies on base64 encoded, serialized protocol buffers
- `.message` files: "backups" in case of crashes

Translation Request Messages

- Each `TranslationRequest` first generates a so called "Translation Request Message"
- TRMs encode request id, source/target languages, source text and (once ready) the final translation
- Each TRM can also have (optional) "packet data", a list of key→value pairs which may encode additional data obtained from the translation worker server

Current State of Things

Supported MT Systems

We have implemented worker servers for:

- Google Translate (all language pairs!)
- Microsoft Translator
- Yahoo! Babelfish
- Lucy RBMT (output includes parse trees!)
- **Moses SMT** → we have a related project

Get the source code!

Source code is freely available from github SOCIAL CODING

- http://github.com/cfedermann/mt-serverland

Includes bug tracker, wiki, documentation. We will be happy to include your code extensions!

Happy branching!

Conclusion

We have implemented a MT server network:

- with central access for users and API calls
- worker servers for many different systems
- flexible object models allow easy extension
- system plays nicely with other frameworks
- open source development envisaged!

Thank you!

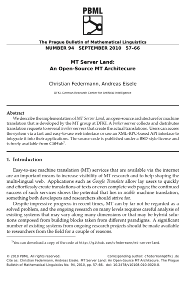
Any questions or comments?!

Publication

Federmann, Eisele. MT Server Land:

An Open-Source MT Architecture.

Prague Bulletin of Mathematical Linguistics, No. 94: pages 57-66, September 2010.



PBML

The Prague Bulletin of Mathematical Linguistics
NUMBER 94 SEPTEMBER 2010 57–66

MT Server Land:
An Open-Source MT Architecure

Christian Federmann, Andreas Eisele

DFKI, German Research Center for Artificial Intelligence

Abstract

We describe the implementation of *MT Server Land*, an open-source architecture for machine translation that is developed by the MT group at DFKI. A *broker* server collects and distributes translation requests to several *worker* servers that create the actual translations. Users can access the system via a fast and easy-to-use web interface or use an XML-RPC-based API interface to integrate it into their applications. The source code is published under a BSD-style license and is freely available from GitHub[1].

1. Introduction

Easy-to-use machine translation (MT) services that are available via the internet are an important means to increase visibility of MT research and to help shaping the multi-lingual web. Applications such as *Google Translate* allow lay users to quickly and effortlessly create translations of texts or even complete web pages; the continued success of such services shows the potential that lies in *usable* machine translation, something both developers and researchers should strive for.

Despite impressive progress in recent times, MT can by far not be regarded as a solved problem, and the ongoing research on many levels requires careful analysis of existing systems that may vary along many dimensions or that may be hybrid solutions composed from building blocks taken from different paradigms. A significant number of existing systems from ongoing research projects should be made available to researchers from the field for a couple of reasons.

[1]You can download a copy of the code at `http://github.com/cfedermann/mt-serverland`.

© 2010 PBML. All rights reserved. Corresponding author: cfedermann@dfki.de
Cite as: Christian Federmann, Andreas Eisele. MT Server Land: An Open-Source MT Architecure. The Prague Bulletin of Mathematical Linguistics No. 94, 2010, pp. 57–66. doi: 10.2478/v10108-010-0020-8.