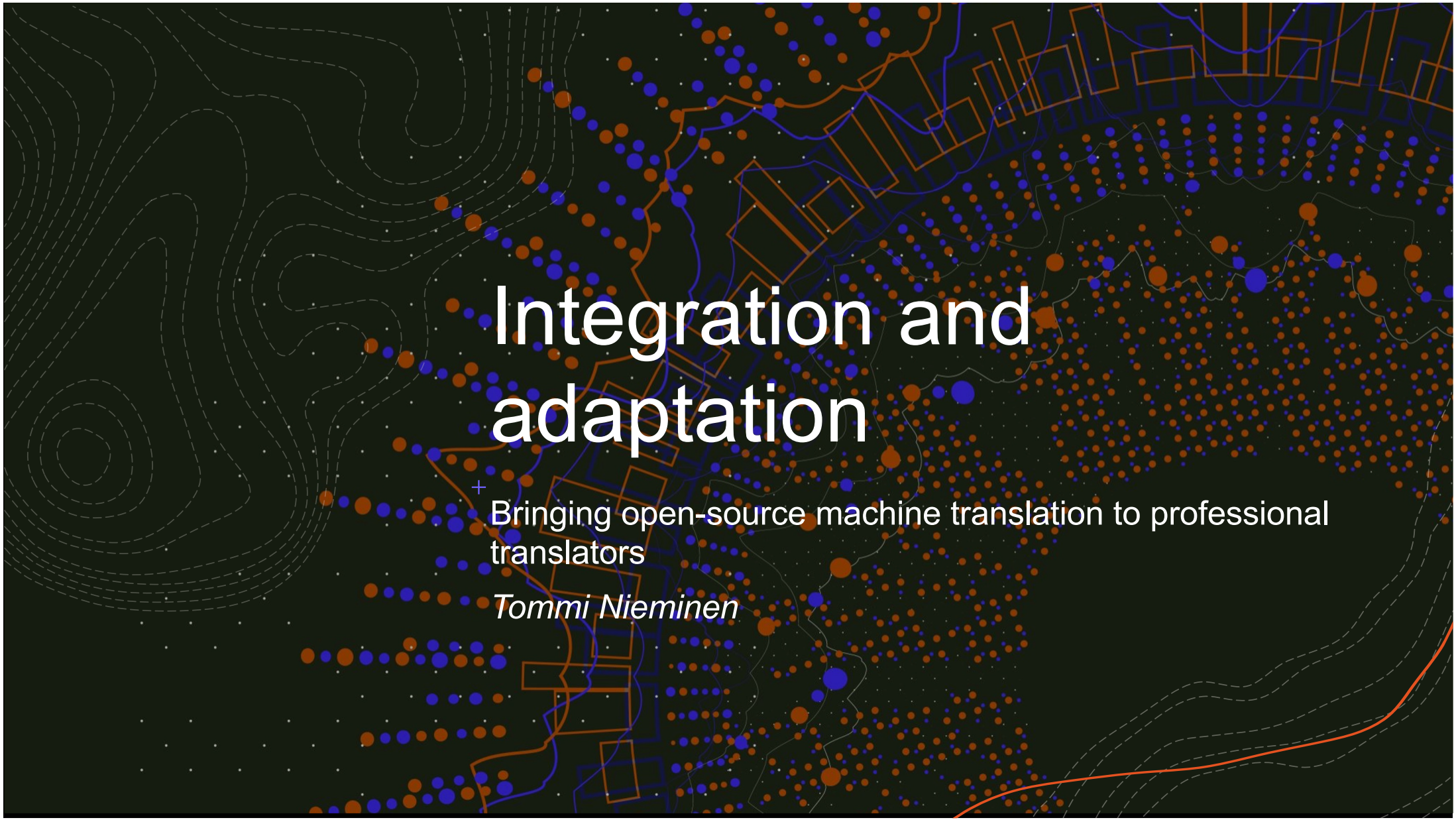

Integration and adaptation
Bringing open-source machine translation to professional translators
Tommi Nieminen

Short CV

Translator (2000-)

- Generalist specializing in games and software translation
- Inhouse, agency and freelancer
- Lecturer on localization, translation tools and machine translation for translators
- Translated the English to Finnish testset for WMT19

Developer (2011-)

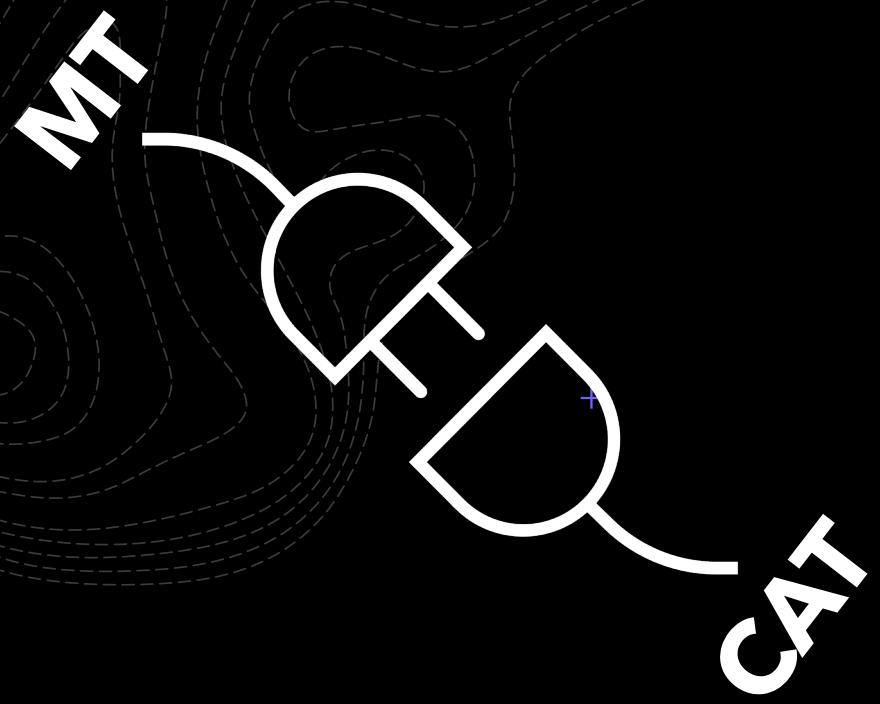
- MA in language technology
- Productivity software for translators (TermInjector, OPUS-CAT), translation workflow automation, machine translation in production use
- Contract work for companies, research work in academic projects (Fiskmö, ELG)

The translation industry

Image credit: Petr Vilgus, licensed under **https://creativecommons.org/licenses/by-sa/3.0/deed.en**

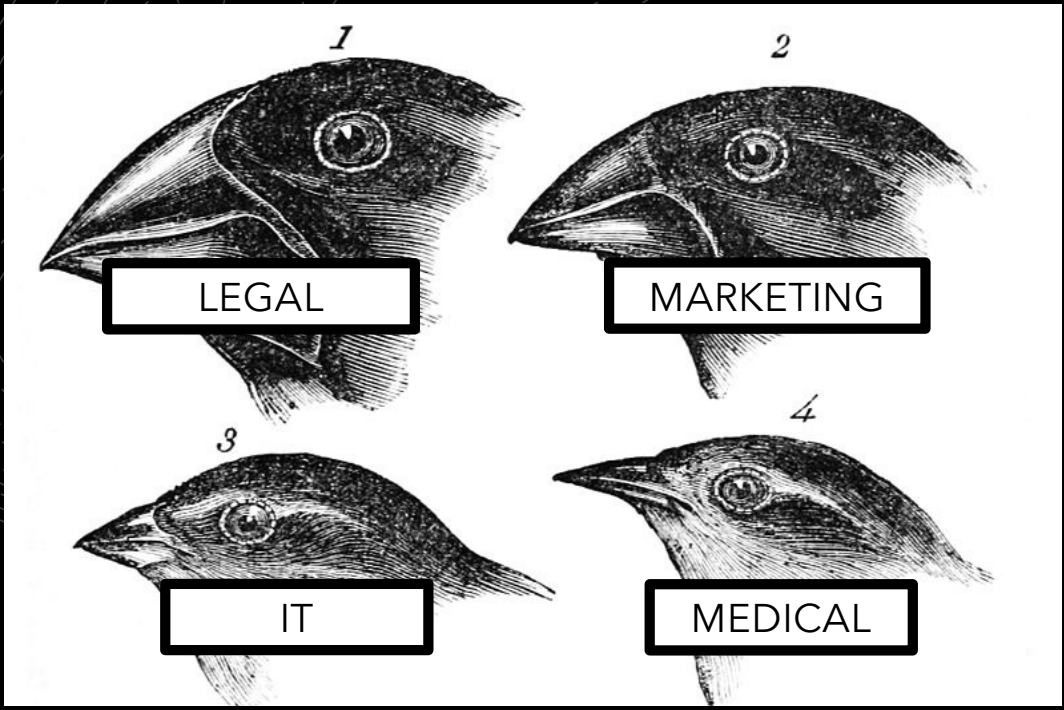
- Early adopter of content generation technology (data normalization, translation memories)
- Abundance of tools and legacy technologies
- Messy collections of legacy language data in form of translation memories and termbases
- Most projects are **brownfield projects**: any new technology must be **integrated** with the existing technology stack and **adapted** to produce content consistent with the legacy data

Integration



- Professional translators usually work in CAT tools
- MT system output must be available inside the CAT tool to be useful
- The latency of making the MT suggestions available to the translator must be as low as possible

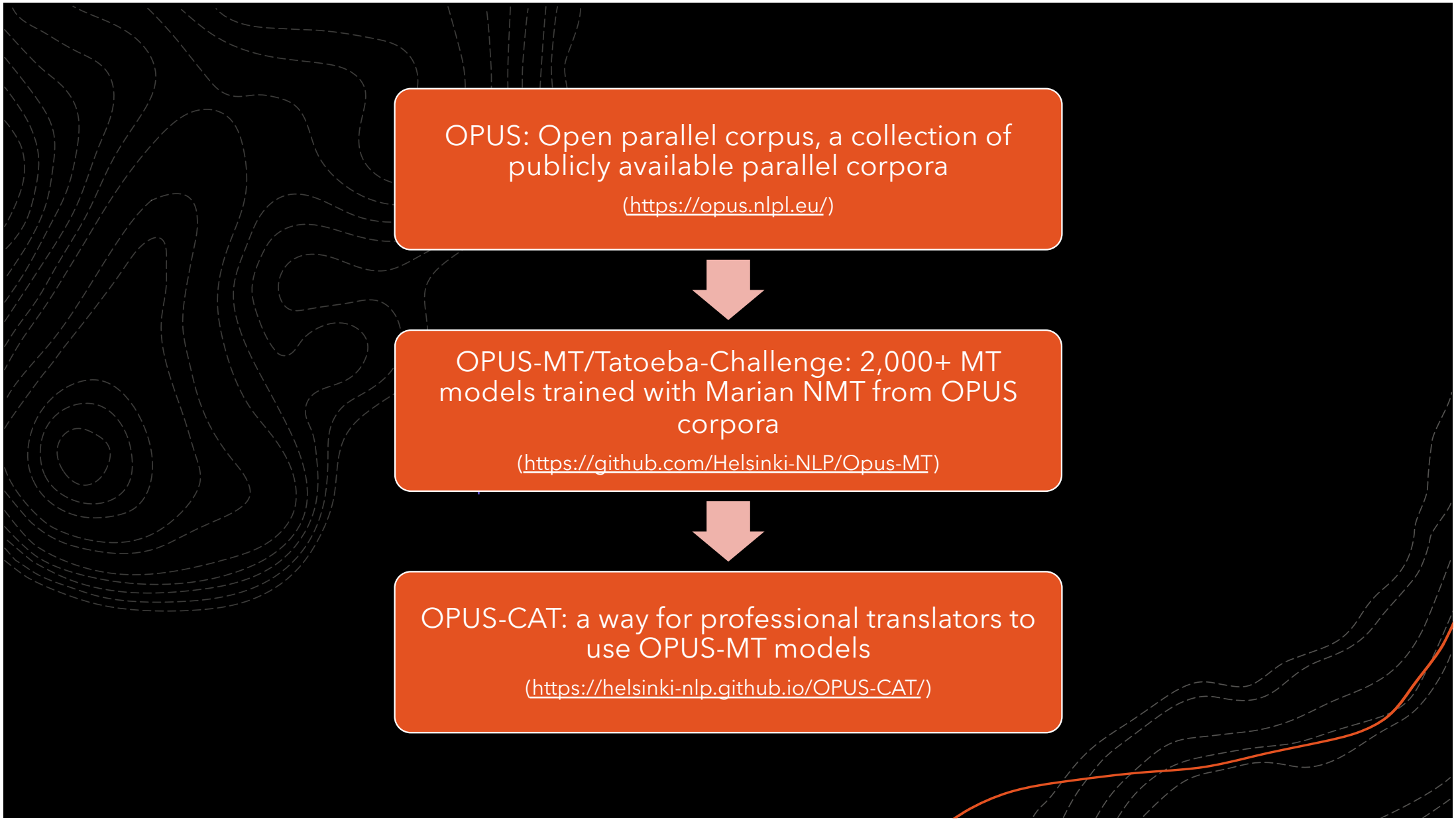
Adaptation

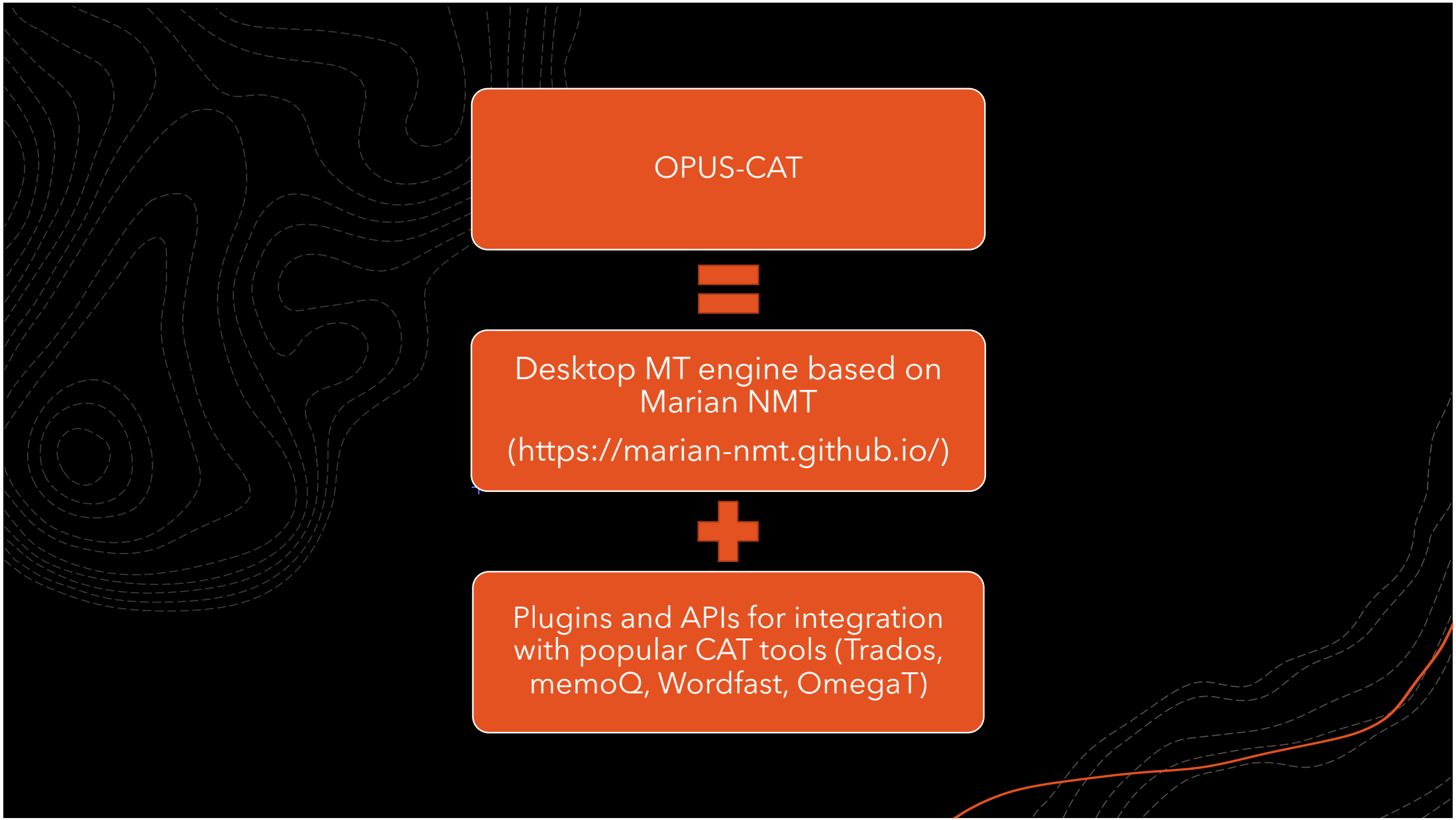


- Most MT models are trained on data from a mix of domains
- Most translation projects have a clearly defined domain (IT, electronics user manuals, legal texts, marketing, medical etc.)
- Generic domain MT is usually not useful in professional translation (too many edits required for publication quality)
- Even domain-specific MT needs to be further adapted for client tone

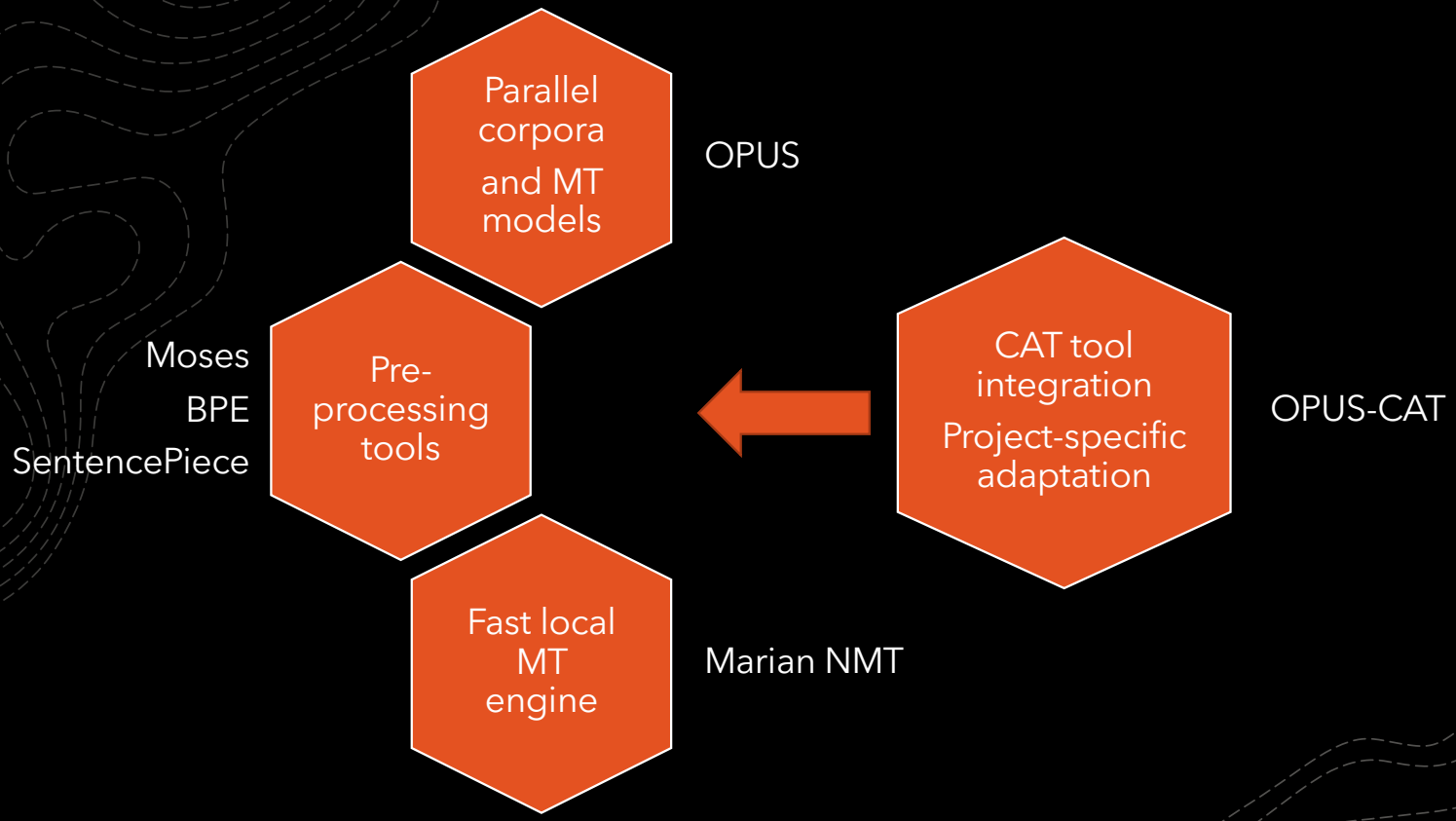
OPUS-CAT

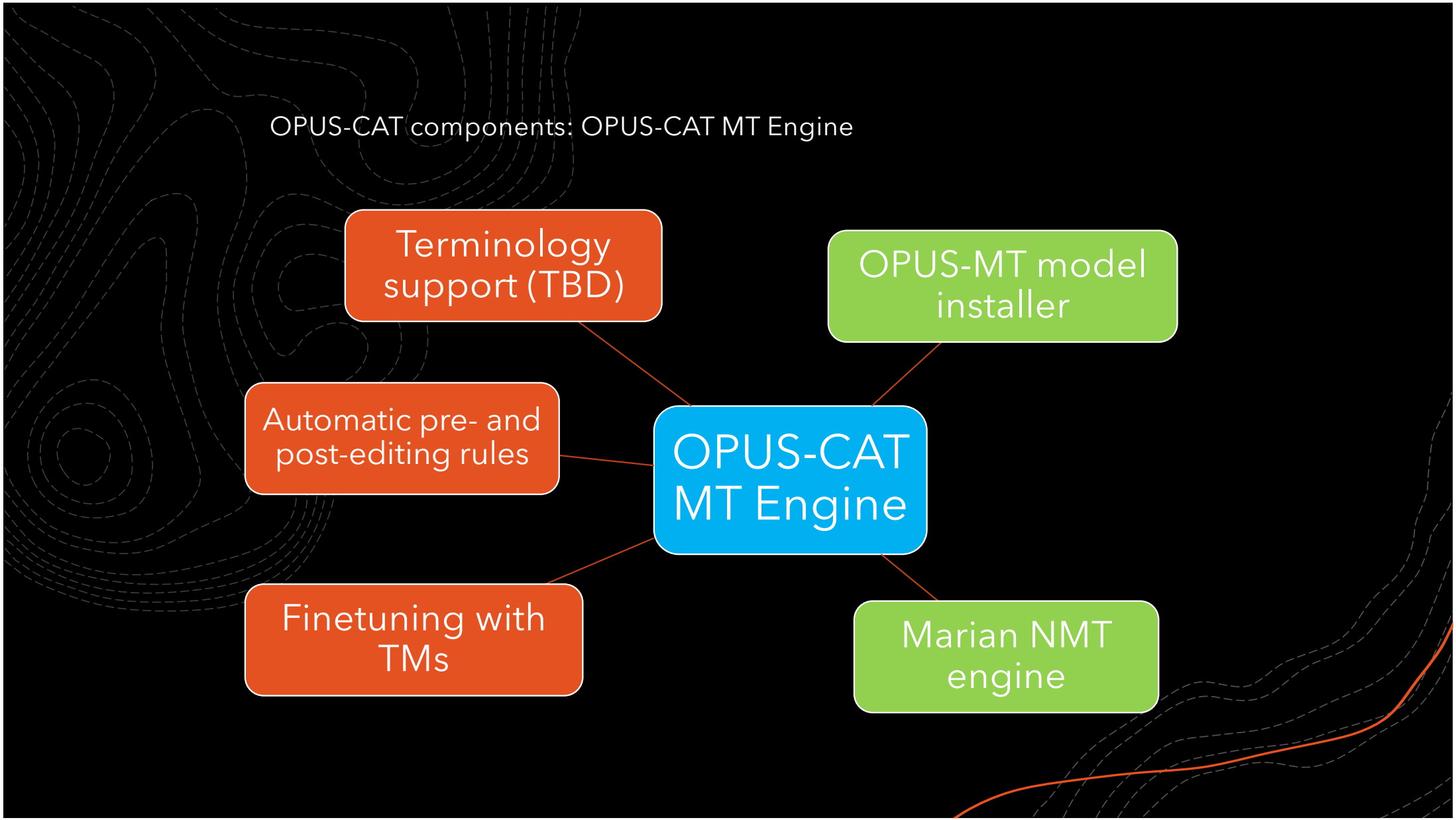
An open-source project, which has been developed in the course of two academic projects, Fiskmö (funded by Svenska kulturfonden) and ELG.


OPUS: Open parallel corpus, a collection of publicly available parallel corpora
(https://opus.nlpl.eu/)
OPUS-MT/Tatoeba-Challenge: 2,000+ MT models trained with Marian NMT from OPUS corpora
(https://github.com/Helsinki-NLP/Opus-MT)
OPUS-CAT: a way for professional translators to use OPUS-MT models
(https://helsinki-nlp.github.io/OPUS-CAT/)

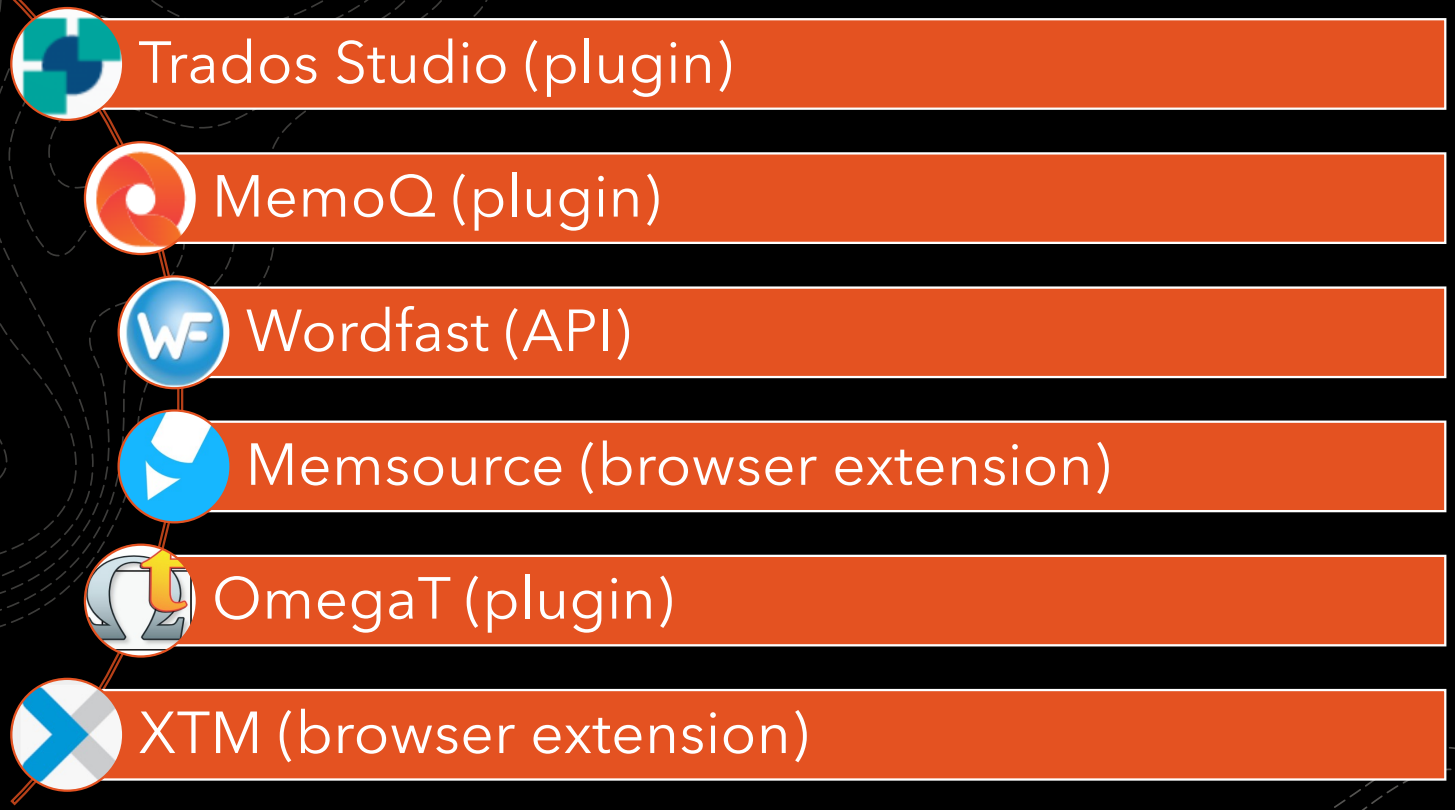

OPUS-CAT
Desktop MT engine based on Marian NMT
(https://marian-nmt.github.io/)
Plugins and APIs for integration with popular CAT tools (Trados, memoQ, Wordfast, OmegaT)

OPUS-CAT: Missing piece of the open source puzzle for MT in professional translation




OPUS-CAT components: OPUS-CAT MT Engine
Terminology support (TBD)
OPUS-MT model installer
Automatic pre- and post-editing rules
OPUS-CAT MT Engine
Finetuning with TMs
Marian NMT engine

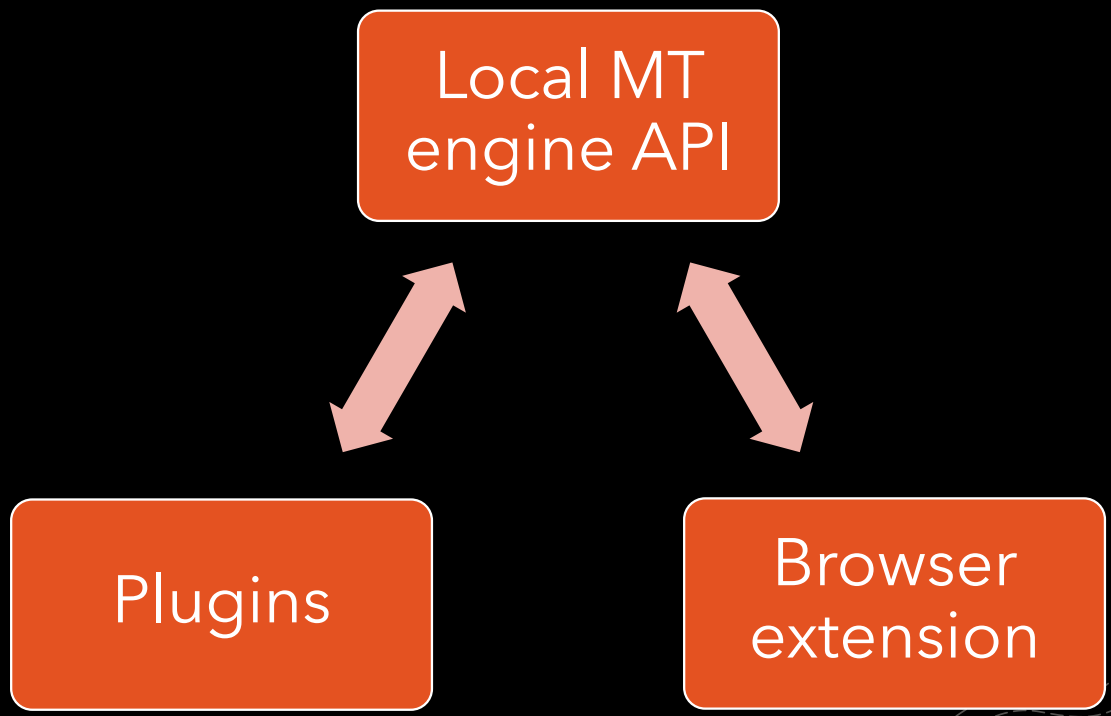
OPUS-CAT components: CAT tool integration



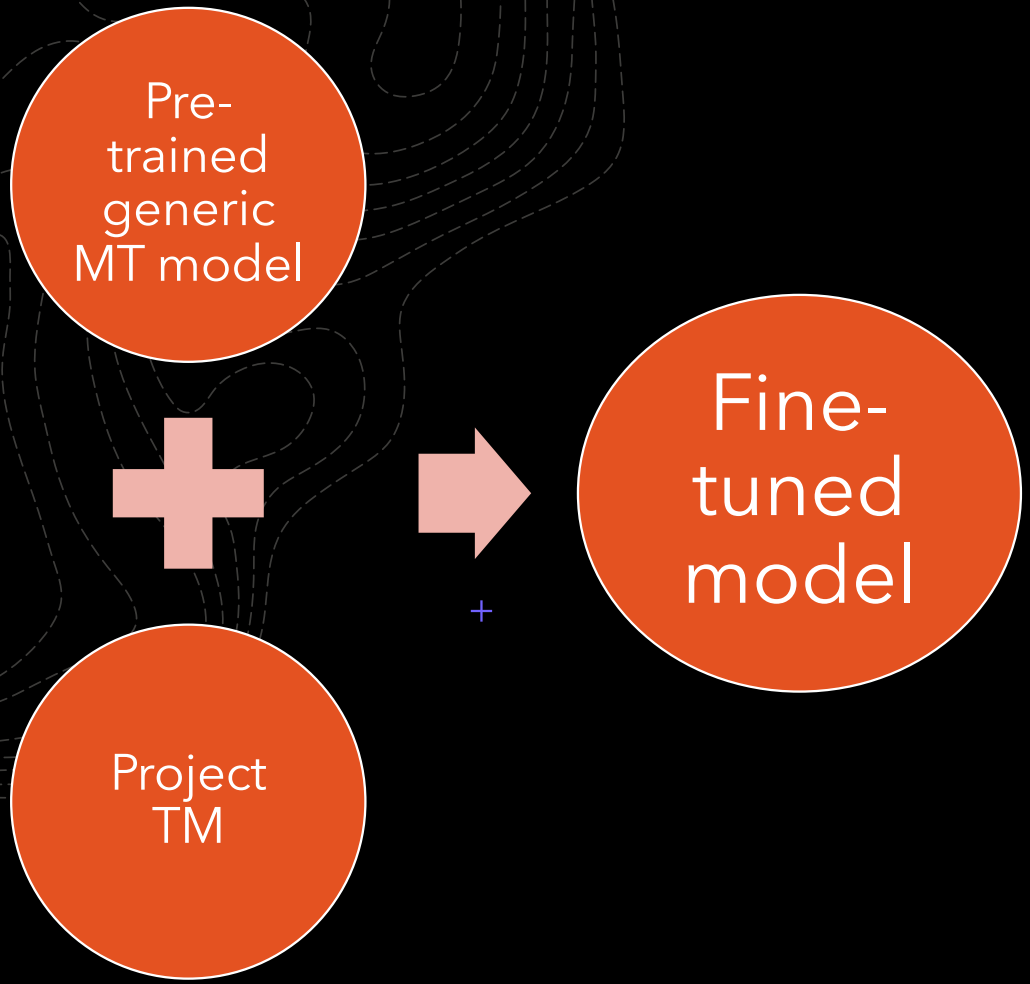
OPUS-CAT components: CAT tool integration

All plugins and integrations communicate with an API exposed by the local MT engine

The API mostly serves translations on request, since most CAT tools do not allow for more complex actions in plugins (except for Trados).



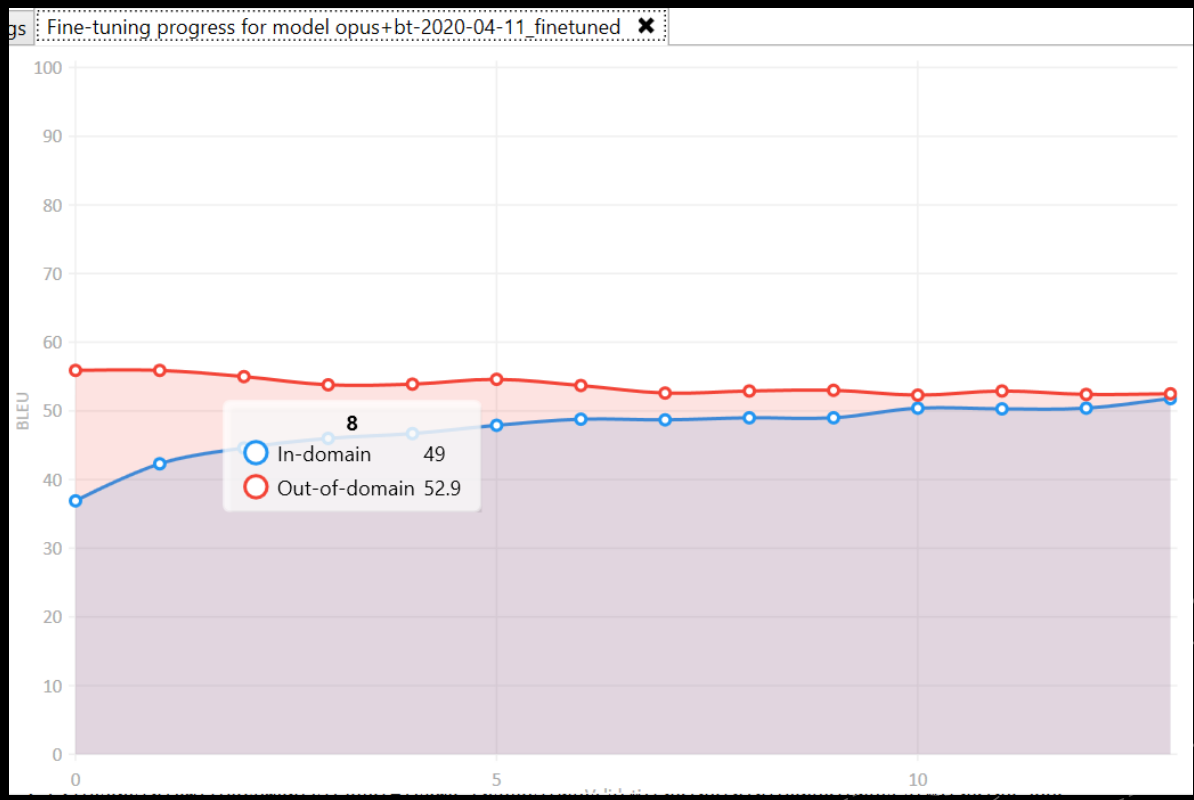
Adaptation in OPUS-CAT: Fine-tuning



Fine-tuning is performed locally on the translator's computer.

Adaptation in OPUS-CAT: Fine-tuning

- Fine-tuning simply means training the model with a fine-tuning set for one epoch (with a small learning rate to prevent quality degradation).
- Fine-tuning has a noticeable effect even with fairly small fine-tuning sets (e.g 10,000 sentence pairs).
- Fine-tuning duration is usually measured in hours (running on a single CPU core), and the model is periodiocally evaluated against both in-domain and out-of-domain test sets.



Adaptation in OPUS-CAT: Fine-tuning

- Even though fine-tuning is an advanced feature, it seems to be popular among users. Examples:
  - a workshop on using the feature in translator conference
  - an MA thesis on using the fine-tuning at Suomen Pankki.
- The effectiveness of the fine-tuning procedure has also been recently manually evaluated (see table on the right).

Human Evaluation Results: Ranking

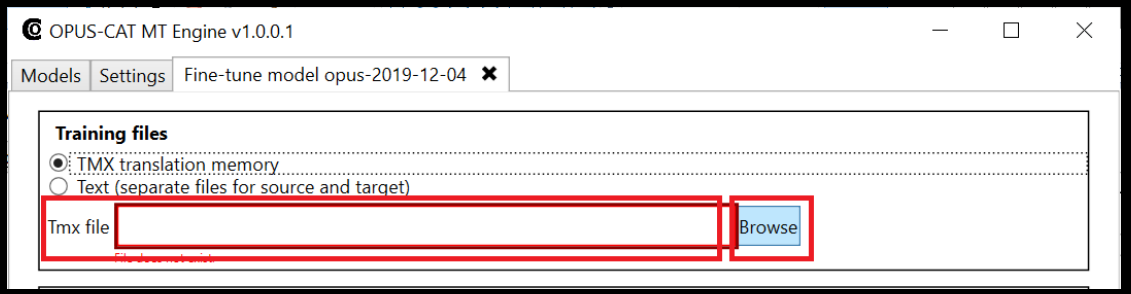
Ranking (%)	Baseline	Small Corpus	Big Corpus
EN → TR	67.57	77.94	**80.63**
EN → ES	75.03	77.25	**80.21**
EN → CA	67.72	72.38	**79.05**

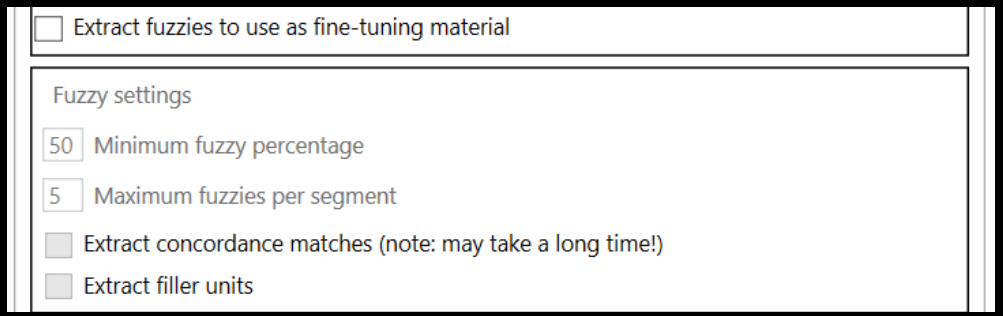
Image credit: Gökhan Doğru. Source: https://asling.org/tc44/slides/TC44-Dogru-A%20User%20Study%20on%20Machine%20Translation%20Fine-tuning%20with%20Custom%20Translation%20Memories%20in%203%20Language%20Pairs.pdf

Adaptation in OPUS-CAT: Fine-tuning

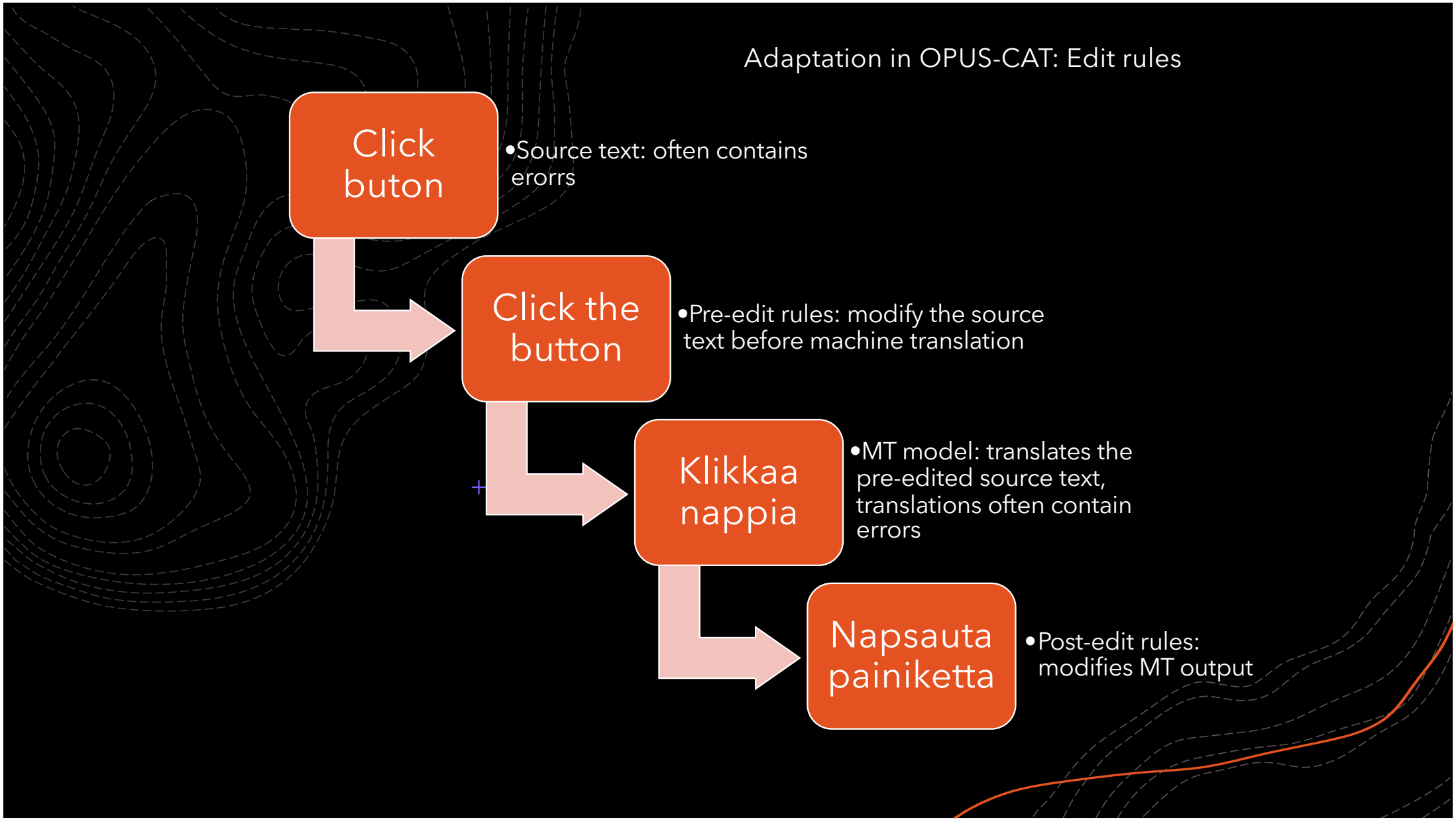
Potential improvements to the fine-tuning feature:

1. The procedure of fine-tuning has manual steps that could be automated (TMX conversion).
2. Fine-tuning should preferably happen in real time when the translator translates (adaptive NMT).
3. In cases where there is no existing TM, it could be useful to fine-tune the base model with data extracted from open corpora (based on source sentence similarity).

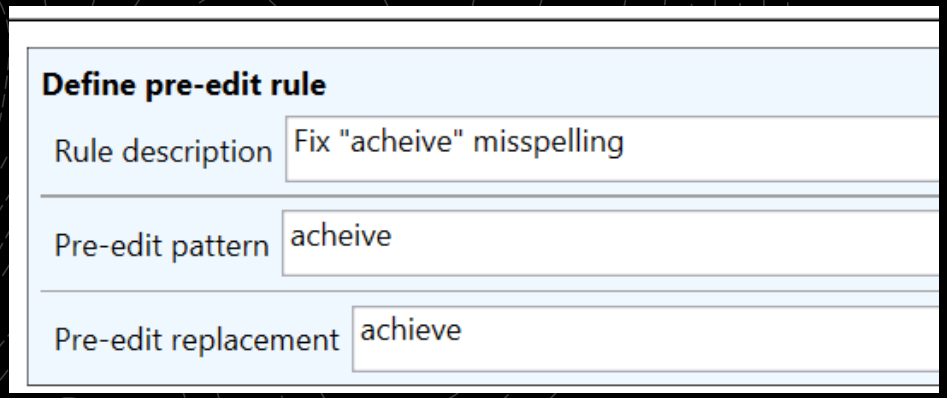


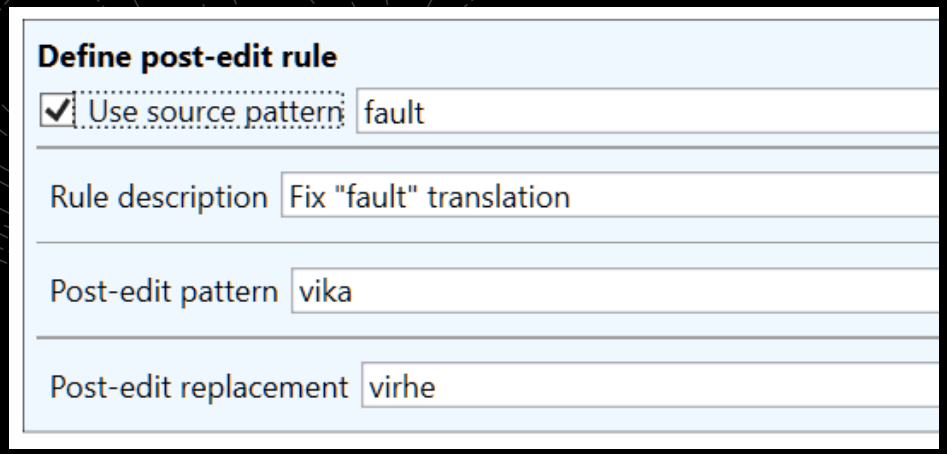


Adaptation in OPUS-CAT: Edit rules



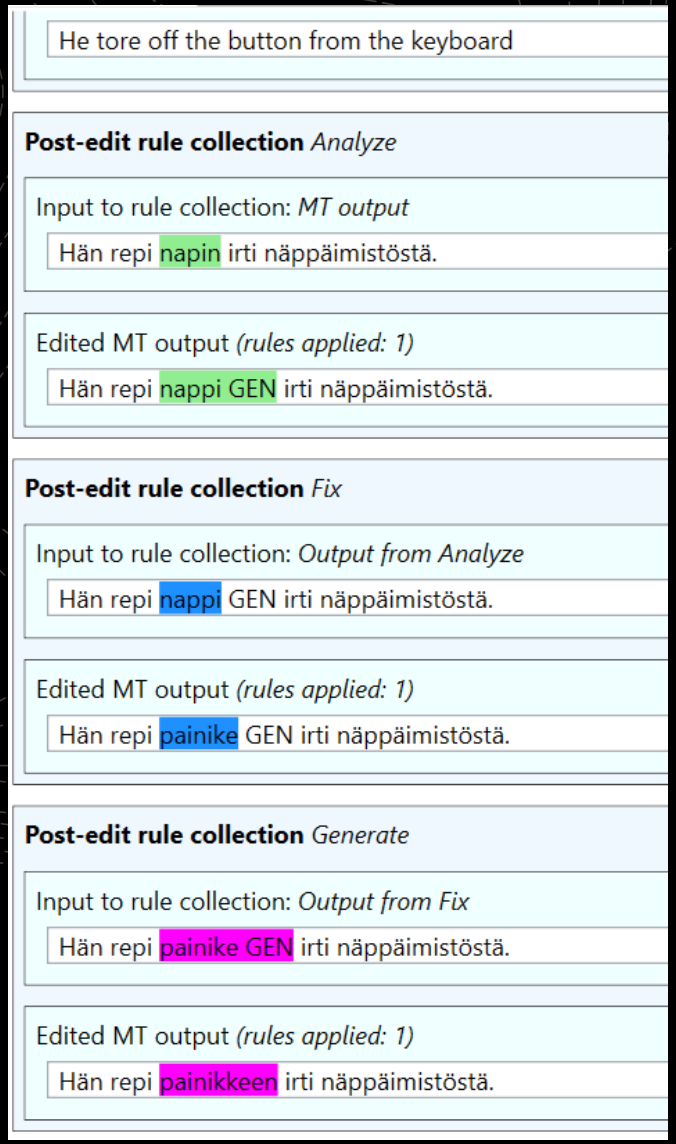
Adaptation in OPUS-CAT: Edit rules





- Edit rules are simple replacement rules (with regexes or not).
- Pre-edit rules edit input before translation (upper left image).
- Post-edit rules edit MT output (they can be made conditional on the source text, and they can also capture text from the source).

Adaptation in OPUS-CAT: Edit rules



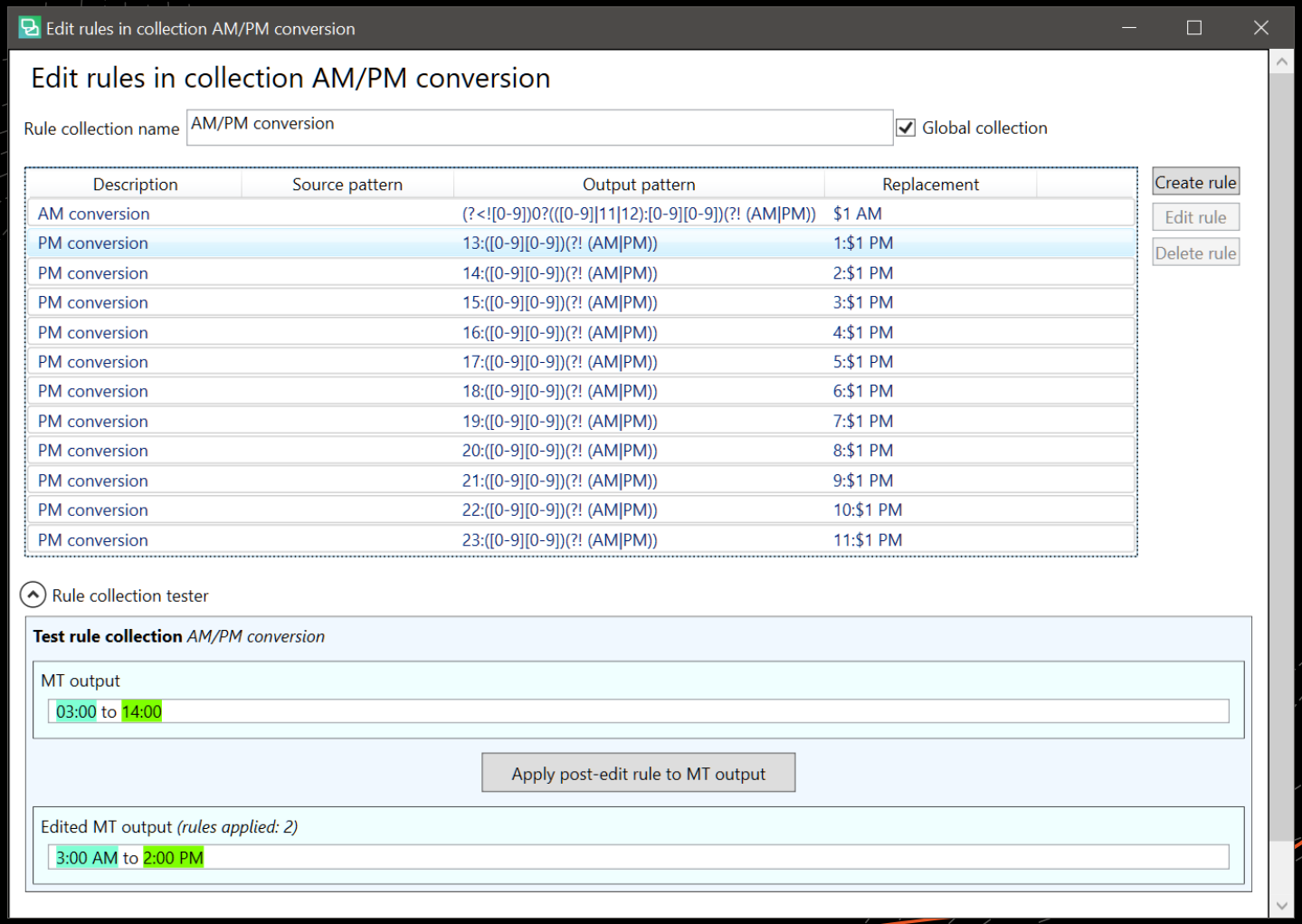
- Complex changes can be implemented by chaining rules so that each rule in the chain takes the output of the previous rule as its input.
- The MT engine contains a rule testing functionality to make it easier to develop rule collections (see image on the left).

Adaptation in OPUS-CAT: Edit rules

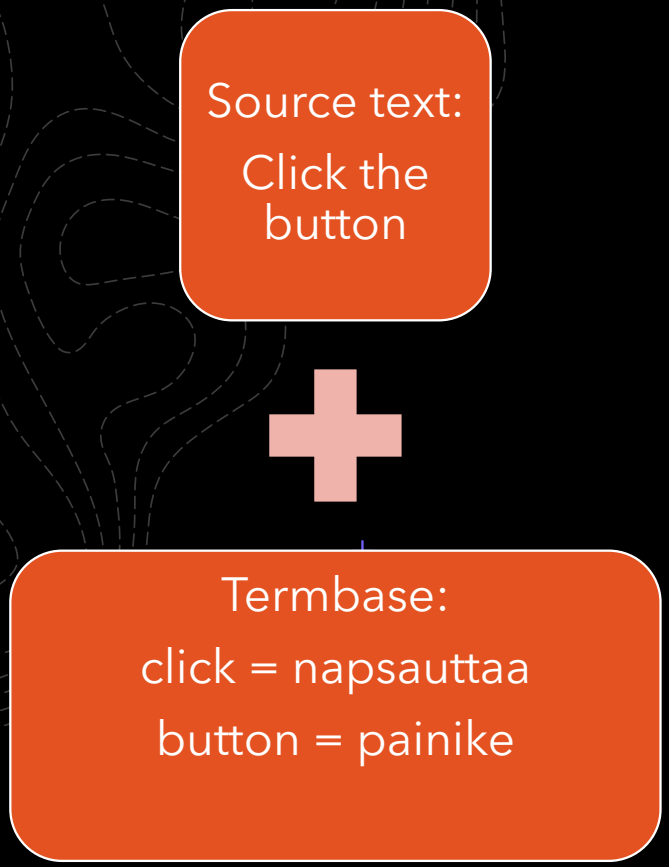
Real-life example: correcting incorrect time formats on MT output (note that the rules can be quickly added and edited in .yml files)

Potential improvements to the rule feature:

1. Alignment-based edit rules.
2. Black-list rules for the NMT decoder.



Adaptation in OPUS-CAT: Terminology support (work in progress)



Napsauta painiketta

Note that terms need to be inflected.

Adaptation in OPUS-CAT: Terminology support (work in progress)

- Terminology support is implemented in OPUS-CAT using the soft constraint method first discussed in Dinu et al. (https://aclanthology.org/P19-1294/).
- Specifically, OPUS-CAT uses a method where alignments are used to find corresponding phrases from the source and target sentences (simulating terminology), and the lemma form of the target phrase is then added to the source sentence after the corresponding source phrase.
- When translating, the target lemmas of recognized source terms are added to the source sentence before decoding.

+

Original source: ▁Exhaust ▁emissions
Target: ▁Pa ko kaasu päästöt

Augmented source (note that the target lemma has been resegmented):
<term_start> ▁Exhaust ▁emissions <term_end> ▁pakokaasu päästö <trans_end>

Adaptation in OPUS-CAT: Terminology support (work in progress)

- The problem with the soft terminology constraint method is that it requires training new models with terminology support for each language pair
- It may be possible to fine-tune the models locally to support soft constraints, a similar method is already used for tag restoration.
- The term support feature still requires a lot of testing and evaluation before it can be officially included in OPUS-CAT.

OPUS-CAT is an open source, completely free alternative for anyone wanting to use NMT in professional translation.

Thanks to Marian NMT, it is possible to generate translations quickly and even fine-tune models with domain-specific data, even on normal Windows desktop computers.

The boxes below describe how OPUS-CAT addresses four factors, which are essential to using MT in professional translation.

Confidentiality	+	Cost-effectiveness	Customizability	Reliability
• Guaranteed confidentiality: no connections to outside services required.		• Completely free to use: OPUS-CAT project has minimal running costs.	• Models can be fine-tuned easily. Large-scale customization is also possible (with GPUs).	• No risk of service outages or closures: desktop MT will work for as long as it is needed.

OPUS-CAT is used by many independent translators and organizations. There is plenty of scope for growth, though, since only a small portion of translators use MT regularly (European Language Industry Survey **2022**).

In addition to use by translators, OPUS-CAT is also useful for researchers and developers.

Testbed for new translation technology	Platform for user studies	Open source components for developers
• New features can be quickly prototyped in production environments by building on the OPUS-CAT codebase.	• OPUS-CAT can be used to study how translators work. Additional monitoring functionalities can be added freely.	• Developers can modify parts of OPUS-CAT (e.g. the plugins) and use them in their own products.