

Curb impacts of AI on programming to maintain EU sovereignty

Let's **assume** that LLMs **actually speed up programming** and that the EU cannot catch up to AI coding tools from the US and China. If this is true, what can we do in the EU to **close the gap** enough that efficiency gains from LLMs stay lower than their compute cost?

Contents

1	Practical steps for sovereignty	1
2	Why we need these steps	2
3	How to do it?	2

1 Practical steps for sovereignty

Practical steps at moderate cost:¹

- Enhance copleft **compiler error messages** (work on *gcc*?)
- Create more **linters** (*SonarQube Rules* integration?)
- Package and improve searchable **documentation** (via *texinfo*?)²
- Spread **discoverability of documentation** akin to IntelliJ for Java and [Emacs](#) for elisp, but for most languages, frameworks, and libraries (via *texinfo-lookup*?).
- Provide **templates for infrastructure setup** (with *proxmox*?) and strengthen security ergonomics (with [BSI Grundschutz](#)?).

¹compared to the 1 trillion VC funding for US-based AI companies and who knows how much for Chinese ones.

²Improve search in documentation and add more examples; collect and integrate working examples in primary documentation. Piecing together info is where most people give up. – *thanks to libewa*. Package more documentation as texinfo. Write automatic conversions to texinfo for different documentation tools. Improve man-page integration. Since I started using local info lookup via texinfo for writing Guile Scheme, I wish that were available for every tool.

- Support Free Software **maintainers** and foster infrastructure maintenance in the EU. If you know a project by heart, you can fix it much more easily.
- Work towards more **standardized interfaces** between software components. Maintain reliable **interface generation** from specifications for different languages. Build interface contracts as the common denominator between different services, extendable for specifics. – *Elias Probst*³
- **De-bloat** our stacks and opensource frameworks and common development practices to reduce overhead. – *Elias Probst*
- Improve **responsiveness**, e.g. reduce keyboard to screen latency. – *mlinksva*

2 Why we need these steps

These steps make software development more resilient against changes due to LLMs.

„What if the [worries by The Morpheus hold true](#) and *the EU cannot become competitive in LLM development*, because USA and China monopolized LLM development skill and infrastructure advantages? Which steps can—cheaply—limit the impact of LLMs so our software development stays competitive?“

I asked this question [in Mastodon](#). To contribute, please answer the question there.

AI may still fail as a bursting bubble and take down large parts of the US economy with it (and more), but if it does not, the EU should be prepared to curb its impact—best by already having taken steps like the ones shown here.

And if the AI bubble does not burst, these steps will still help software development in the EU. Money put into them **never goes to waste** (it also helps the ones who use LLMs for coding, just not as much), so I think that the EU should support these steps with at least 1% of the VC funding AI companies receive.

3 How to do it?

- Increase funding of the [nlnet foundation](#) via EU Horizon 2020 and followups.
- Increase funding of the [Sovereign Tech Agency](#).
- Adhere to and require [Public Money Public Code](#).
- Fund the [Free Software Foundation Europe](#).⁴

³OpenAPI in the web-world is something in the right direction, but I feel like this is also sorely needed on many other levels. – *thanks to Elias Probst*.

⁴As private person you can donate to the FSFE.