Martin Scherpinski

19. November 2025

Using AI

to solve a key a11y problem of maps!

con·terra
locate the future

a11y – Accessibility

- Create an inclusive web for everyone
  - for the visually, motorically, cognitive impaired …

- became more and more important over the past years, even by law

- **A11Y IS NOT "JUST A FEATURE"!**
  - and adding it afterwards means a lot of effort....

Image created with ChatGPT

- Typical Problems:
  - **Keyboard accessibility**: Many interactive elements can only be reached with a mouse or touch.
  - **Semantic structure**: Screen readers can't interpret if elements are built without **ARIA roles or labels**.
  - **Focus and live region**: Dynamic content changes (e.g., AJAX/SPAs) without notifying the screen reader.
  - **Contrast and readability**: Low color contrast and non-scalable text make use difficult for many users.

a11y – Problems with maps

Image created with ChatGPT

- Maps are inherently **visual**, dynamical and intercative
  - Screen readers usually only encounter an **empty** `<div>` or `<canvas>`

- Missing alternatives
  - POIs and layer information are rarely available as accessible lists or tables (**Non-Visual View**)
- Maps have no linear structure
  - Spatial **relationships** are mainly visual (e.g., “north of…”) can roughly be transported with plain text in lists

- Interaction issues
  - Zooming, panning, or clicking markers or popups often isn’t **keyboard-accessible**.
- Dynamic content:
  - Screen readers are not automatically informed when maps move, filters change, or layers switch.

Another approach to a11y: AI

- LLMs (Large Language Models) nowadays offer strong capabilities to generate textual descriptions of images
- **Why not use this for a11y?**

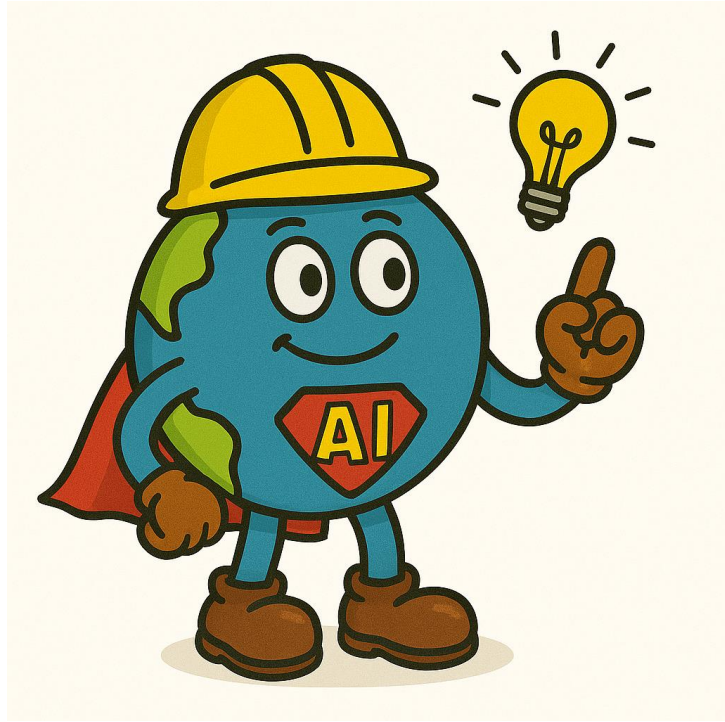


Image created with ChatGPT

- 💡 Concept
  - Users can actively trigger a textual description of the current map view.
  - The application creates a **screenshot**, an LLM analyzes it, and generates a semantic textual description.
  - The description can be read out by a screen reader

- Assumed example output:
  - *“The map shows the city center of Frankfurt with the train station. A river runs along in the south.”*

Advantages of the approach

Image created with ChatGPT

- **Provides access** to dynamic, visually rendered maps (e.g., WebGL, raster maps) that were **previously unreadable** for screen readers.
- **Independent and extensible:** works with any type of layer, map, data type, client
  - Potentially applicable to heatmaps, 3D views, or any thematic maps.
- **Spatial orientation** becomes linguistically perceivable (“north of the city center,” “nearby…”).
- **Visual context** can be included in the analysis like street patterns, map labels, colors of POIs
- **User-controlled interaction**: The user triggers the description on demand, avoiding information overload.

Challenges

- **Accuracy**: AI-generated descriptions may be **incorrect** (“hallucinations”)
  - Output is not **deterministic**, small changes in the input can have different results
  - **Can I trust the output?** ("I think, this could be …")
  - **Context is King!** Best results may be achieved in combination with additional feature data

Image created with ChatGPT

- **Complement, not replacement**: Screenshot descriptions should supplement, not replace, accessible structured alternatives (e.g., ARIA-based POI lists)
- **Latency**: Processing must be fast enough for practical use, screenshot upload needs to be considered

- **Data protection**: Screenshots may contain sensitive (geo-)data
- **Lack of standards**: No WCAG-compliant guidelines **yet** for AI-based descriptions.

Technical realisation – What do you need for a prototype?

- Depends on the target client:
  - Prototyping in **map.apps**, which is based on the ArcGIS Maps SDK for JavaScript

- OpenAI API https://platform.openai.com/docs/api-reference/introduction
  - Rest API to communicate with an LLM like GPT-4.1 (basically, just like ChatGPT)
  - Completions API / Responses API
- API-Key: Directly from OpenAI or from a deployment in Azure Open AI
- Access to a model: le.g. **GPT-4.1** hosted in West-Europe

- Node.js packages: https://www.npmjs.com/package/openai / https://www.npmjs.com/package/@azure/openai
- view.screenshot() from the Maps SDK

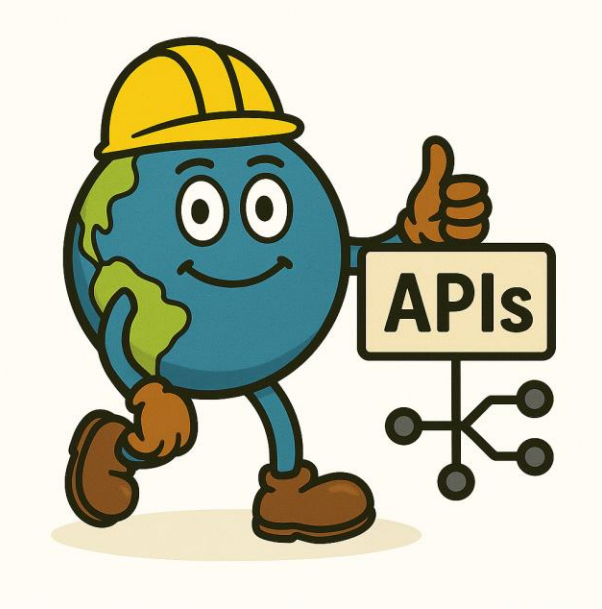


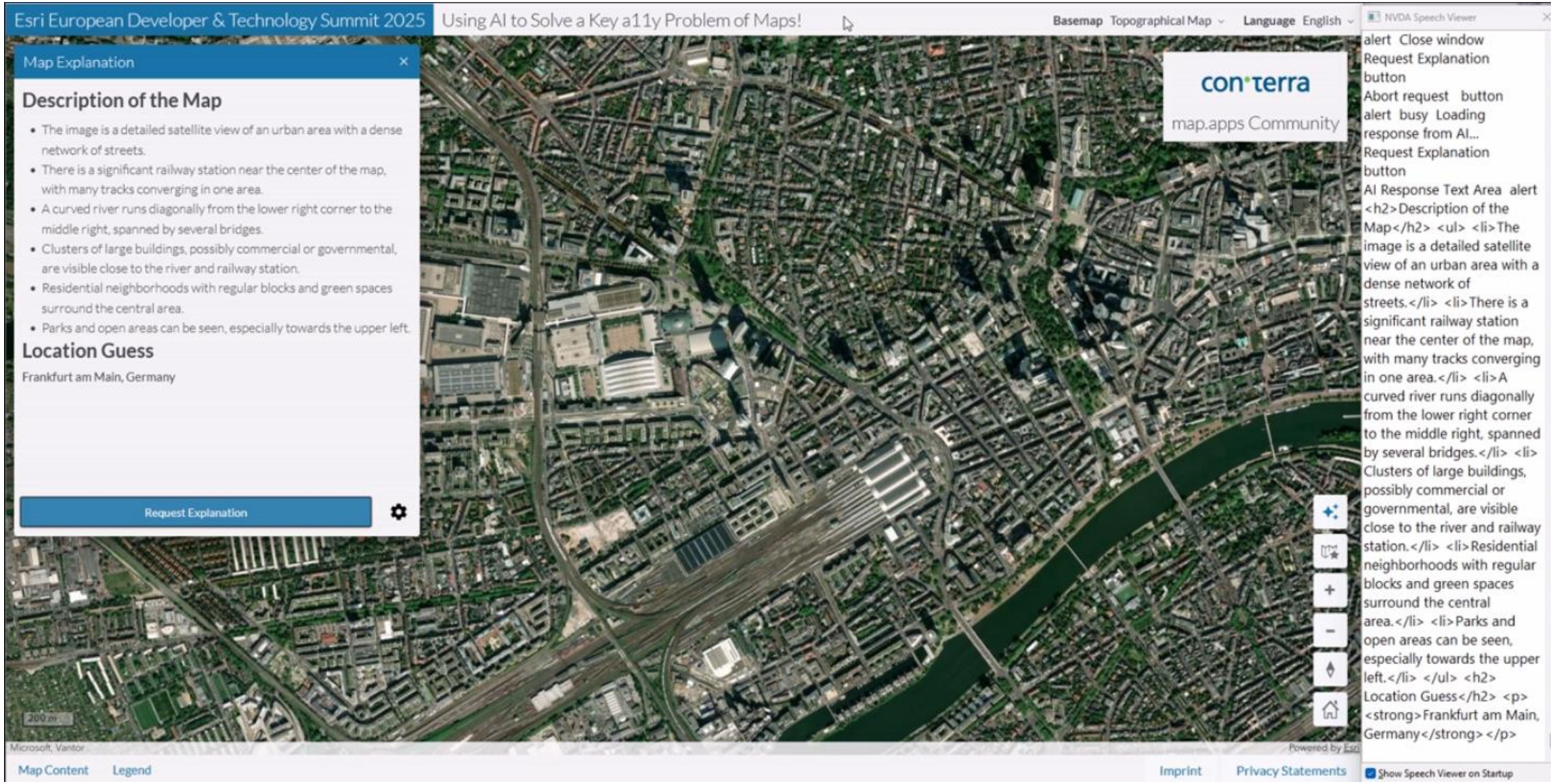
Image created with ChatGPT

Prompting / Interacting with the API

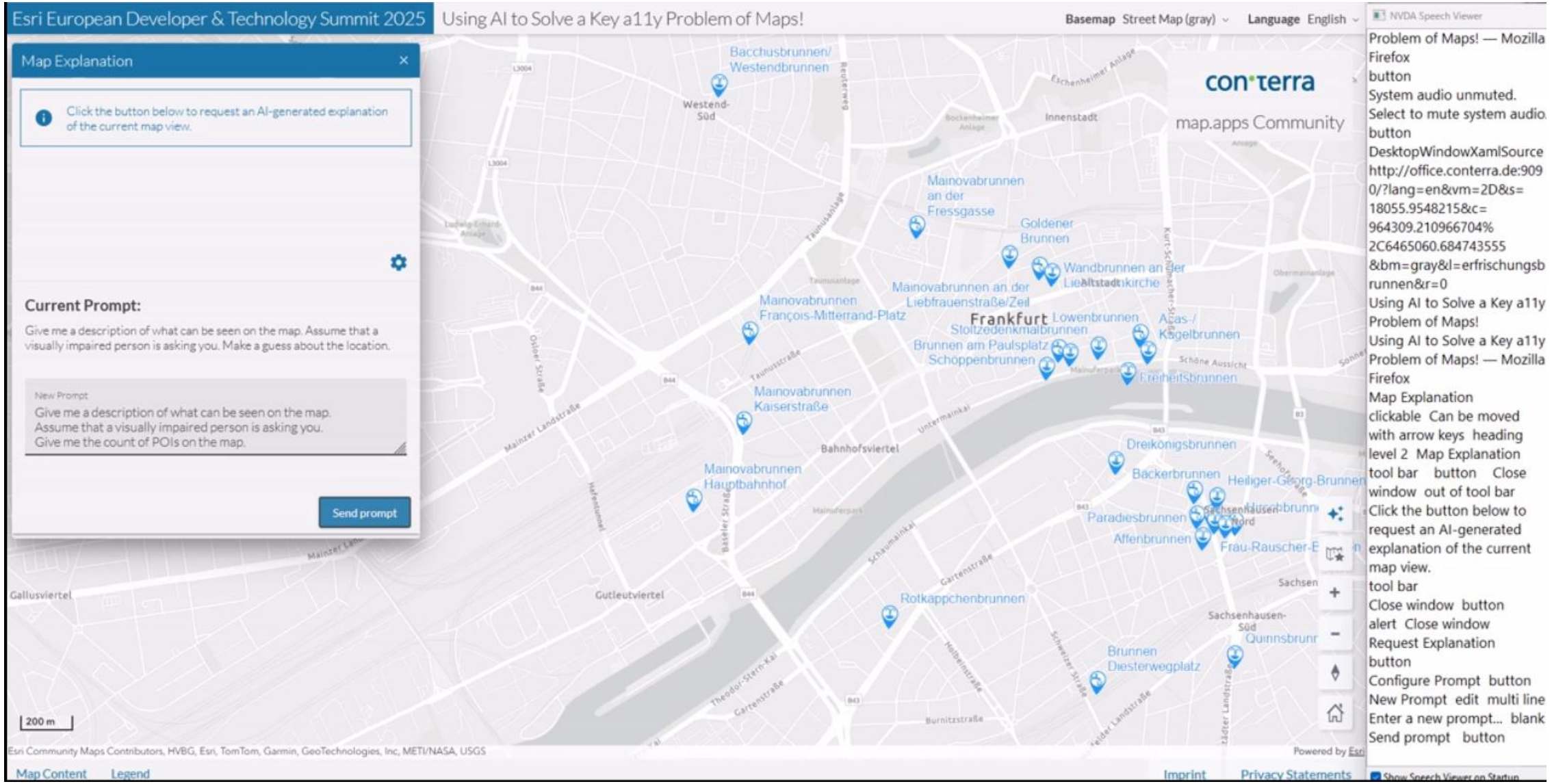
```
{ "messages": [{
            "role": "system",
            "content": [
                {   "type": "text",
                    "text": "Always respond in the language from the locale 'de'.
                             Please provide your answer in visually appealingly formatted html.
                             Make the output accessible, so a screen reader can read it out loud.
                             Do not include markers for the file format in the content.
                             Do not include images in your response."
                }]
        },{
            "role": "user",
            "content": [
                {   "type": "text",
                    "text": "Give me a description of what can be seen on the map.
                            Assume that a visually impaired person is asking you.
                            Make a guess about the location."
                },
                {   "type": "image_url",
                    "image_url": {
                        "url": "..."
                    }
                }]
        }],
    "model": "aoai-gpt-4.1-demo-m.scherpinski",
    "store": true
}
```

Image created with ChatGPT

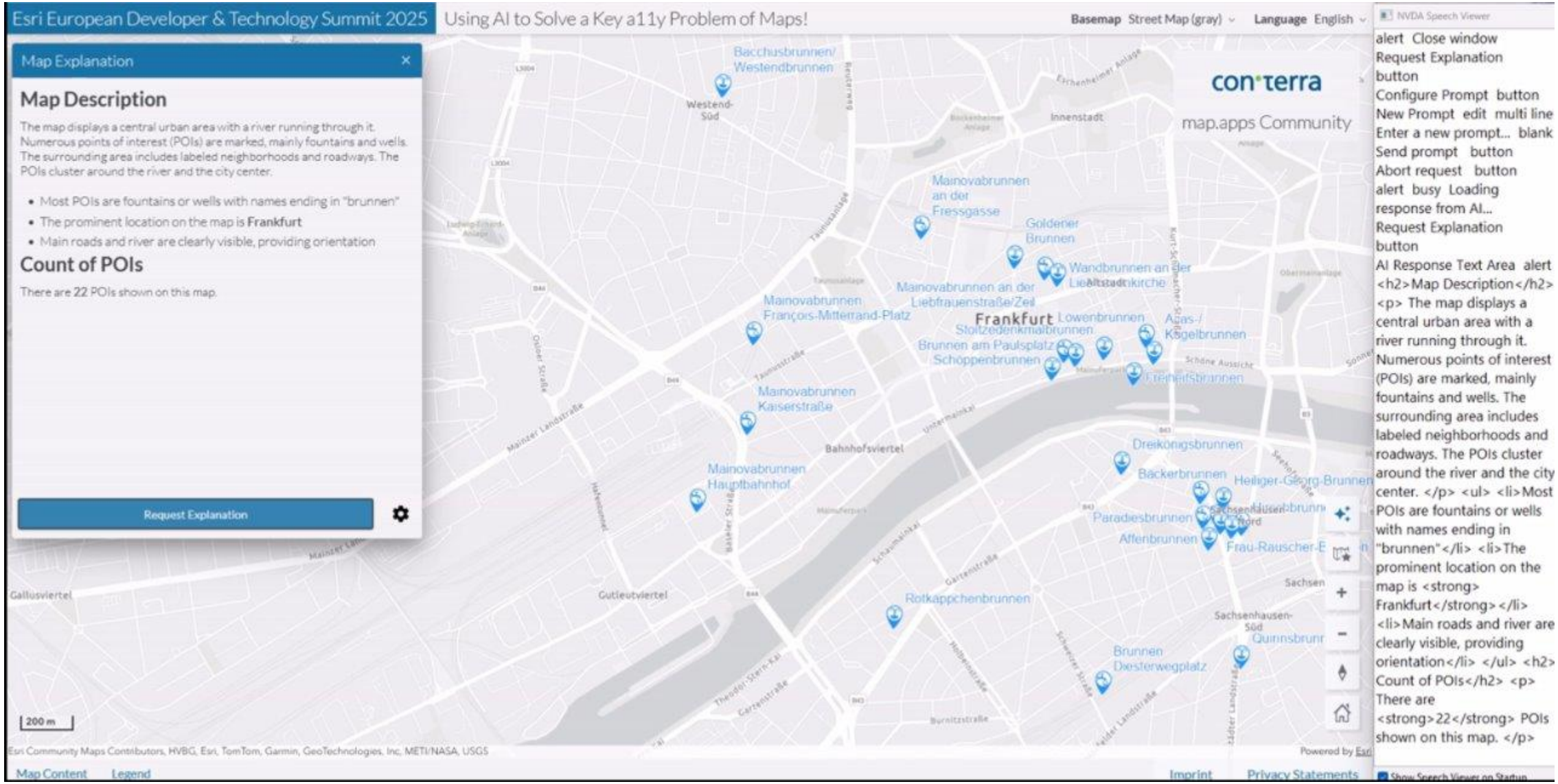
Demo 1 – General workflow with satellite images



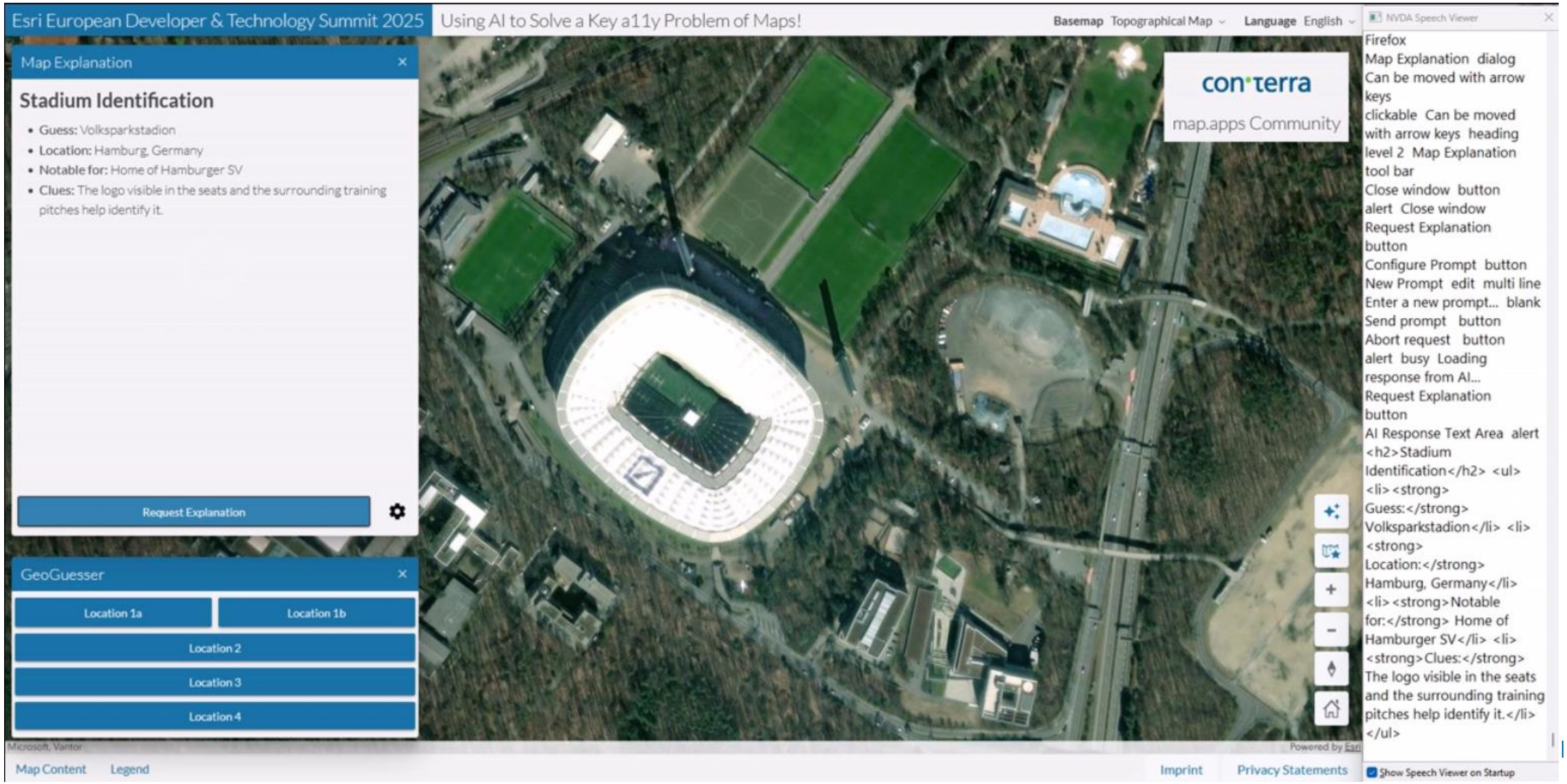
Demo 2 – Different Prompts and Geodata



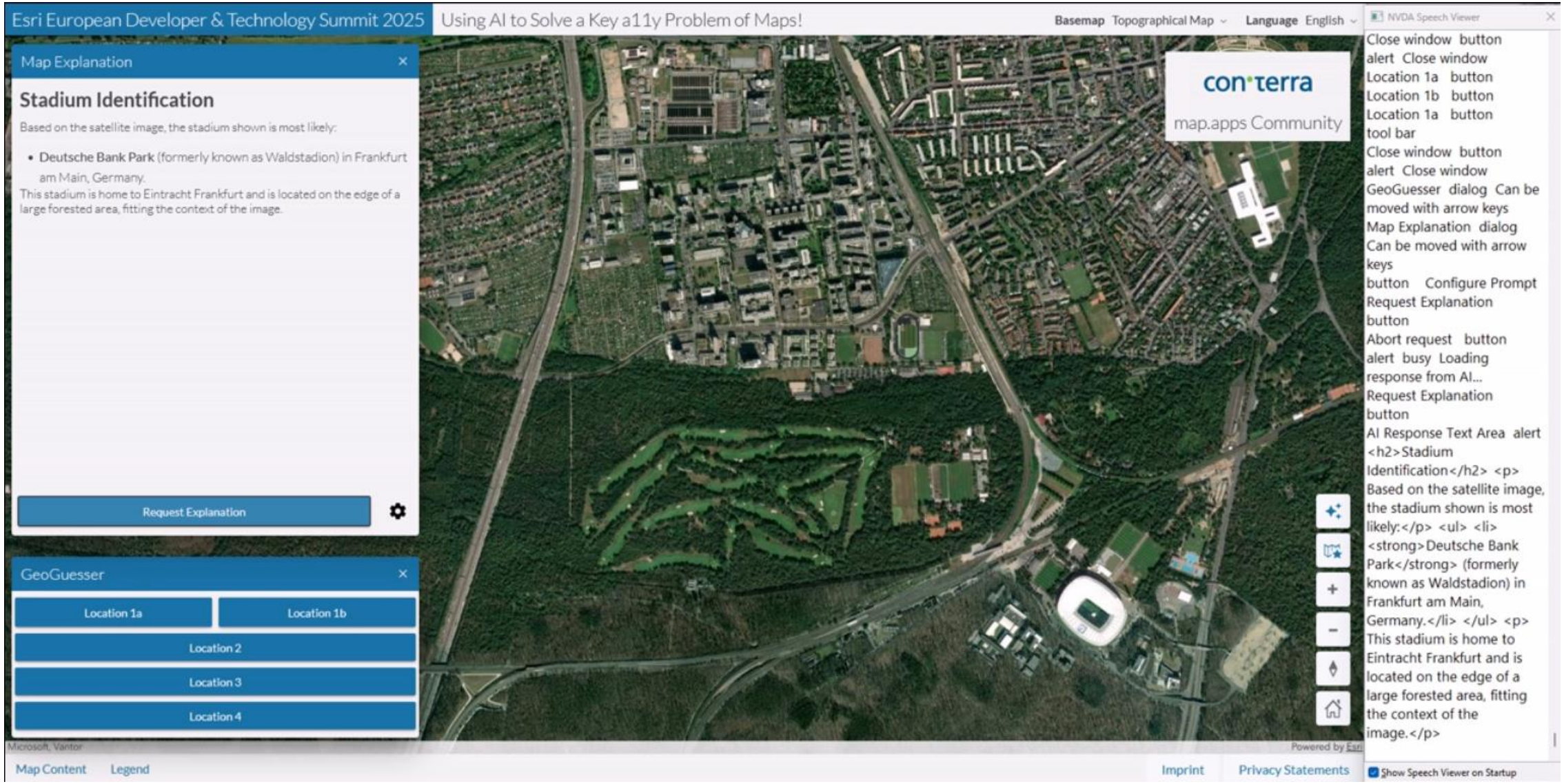
Demo 2 – Different Prompts and Geodata



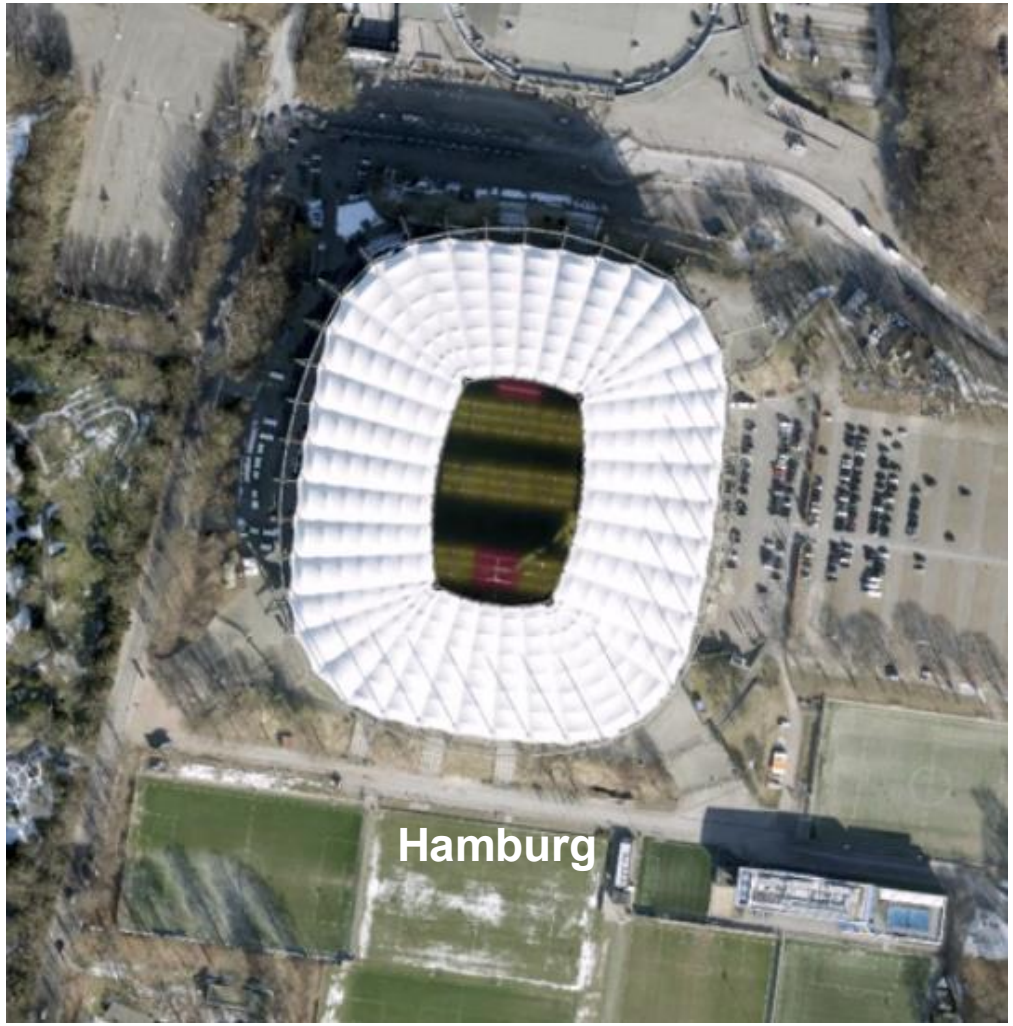
Demo 3 – GeoGuesser and spatial context



Demo 3 – GeoGuesser and spatial context

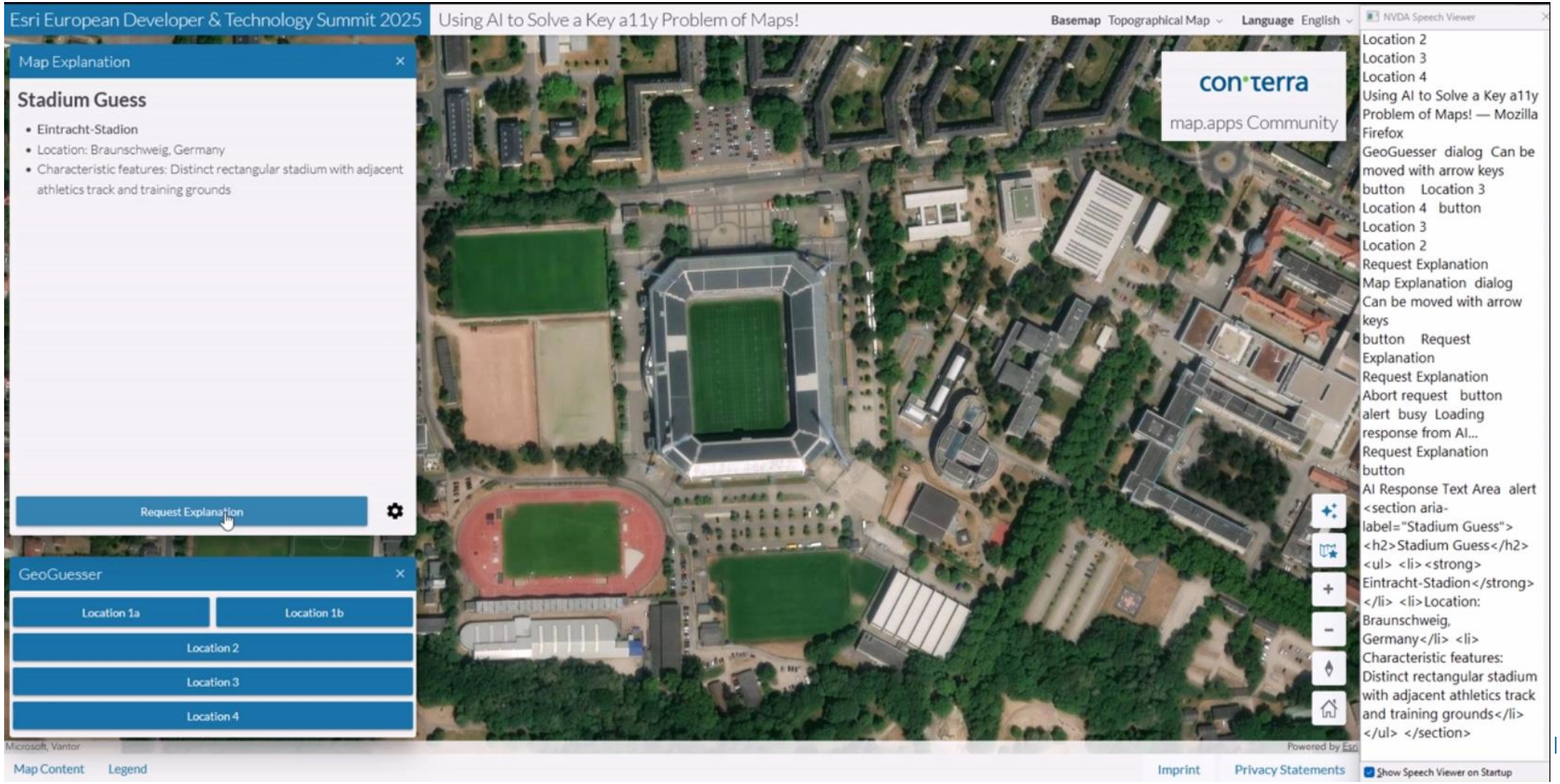


Demo 3 – GeoGuesser and spatial context



Demo 4 – GeoGuesser with more obvious data

Demo 5 – GeoGuesser with labels for context



Demo 5 – GeoGuesser with labels for context

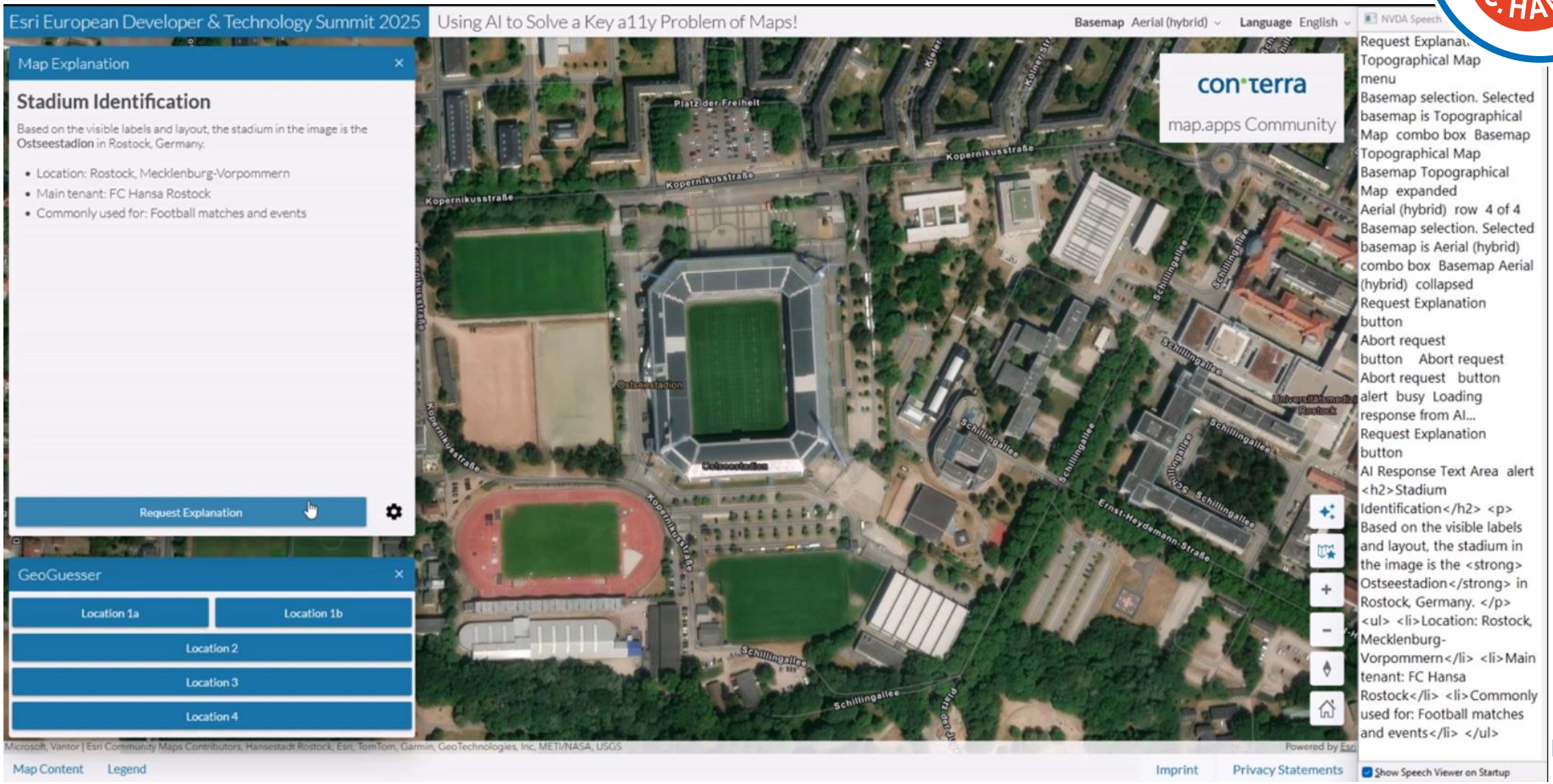


Image created with ChatGPT

Conclusion

- Accessibility for web maps has long been challenging because screen readers cannot interpret spatial structures.

- The **AI approach** with screenshots represents an **innovative bridge**:
  - It translates visual map content into textual, semantic descriptions, making it **understandable for visually impaired users**.

- **Good Practice for map a11y:**
  - **Combine traditional accessibility methods (ARIA, keyboard navigation, structured lists) with an optional AI-powered descriptive layer** for contextual understanding. Use screenshots and actual feature data to boost the results.

Follow us!

linkedin.com/company/con-terra-gmbh

kununu.com/de/con-terra

instagram.com/conterra.gmbh

youtube.com/conterrachannel

con terra Newsletter

conterra.de/aktuelles/newsletter

Thank you very much!

Find me at the expo for questions!

Martin Scherpinski
Software Engineer

con terra
Martin-Luther-King-Weg 20
48155 Muenster
Germany

T +49 251 59689 300
info@conterra.de
con-terra.com

Image created with ChatGPT

Slides
https://github.com/m-scherpi/a11y-ai