

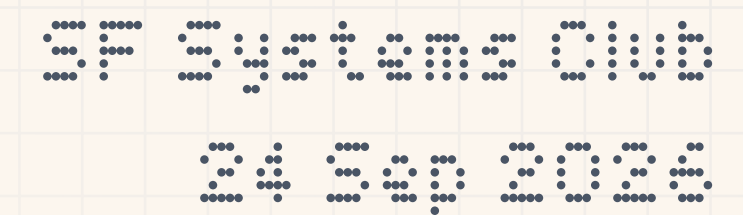
Natural Language

Direct Manipulation

Text Editing



[https://parkie-doo.computer/
peziegler@cs.berkeley.edu](https://parkie-doo.computer/peziegler@cs.berkeley.edu)



RAISE YOUR HAND

When I think about programming,
I think about typing code into



+ your favorite text
editor I missed

■ Text is expressive.

■ Text is portable.

■ Text is analyzable.

Text is great!

■ Text is expressive.

■ Text is portable.

■ Text is analyzable.

Text is great!

THE CAVEAT

But text editing is not without
its warts!

■ Text editing is
error prone.

■ Text editing is
slow.

■ Text editing is (usually)
non-interactive.

THE CAVEAT

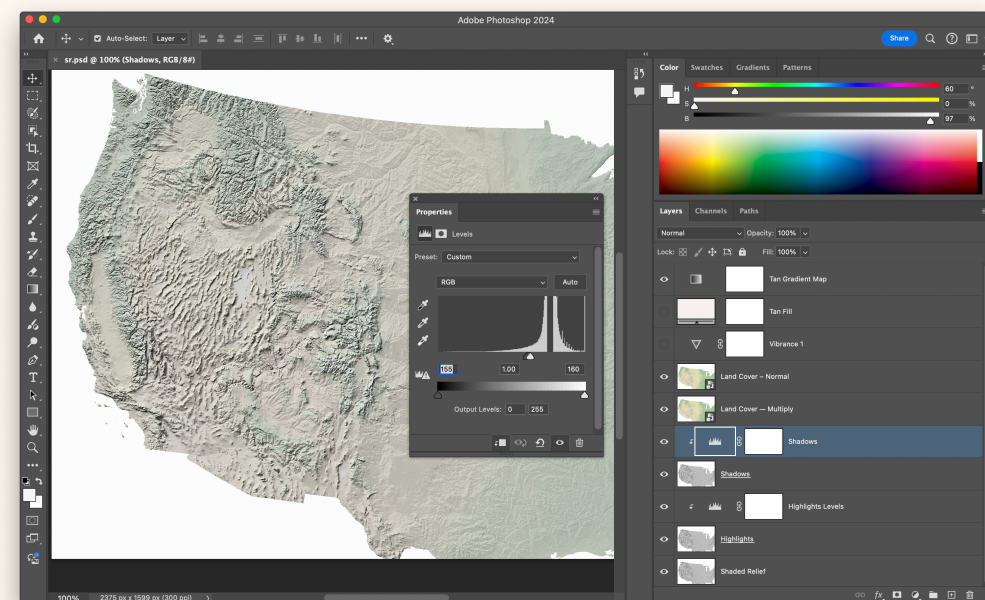
But text editing is not without its warts!

■ Text editing is error prone.

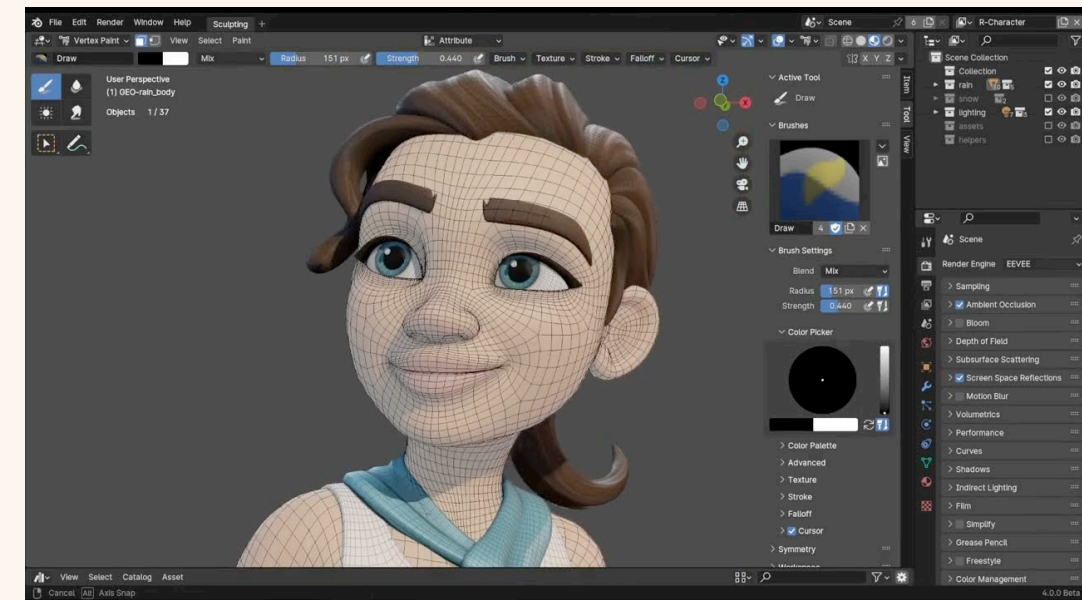
■ Text editing is slow.

■ Text editing is (usually) non-interactive.

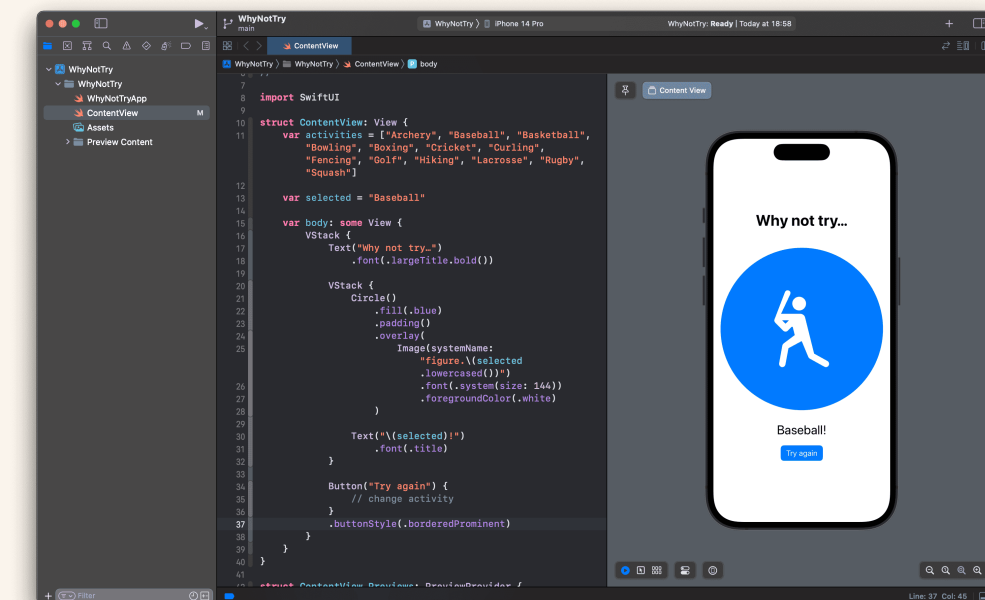
IMAGE PROCESSING



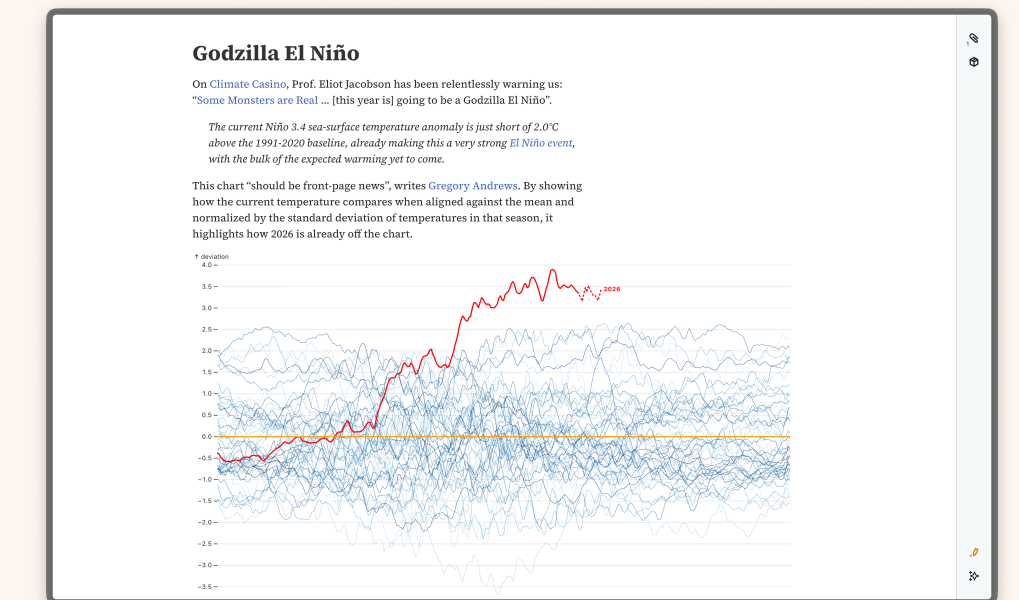
ANIMATION AND MODELING



MOBILE DEV



DATA VISUALIZATION

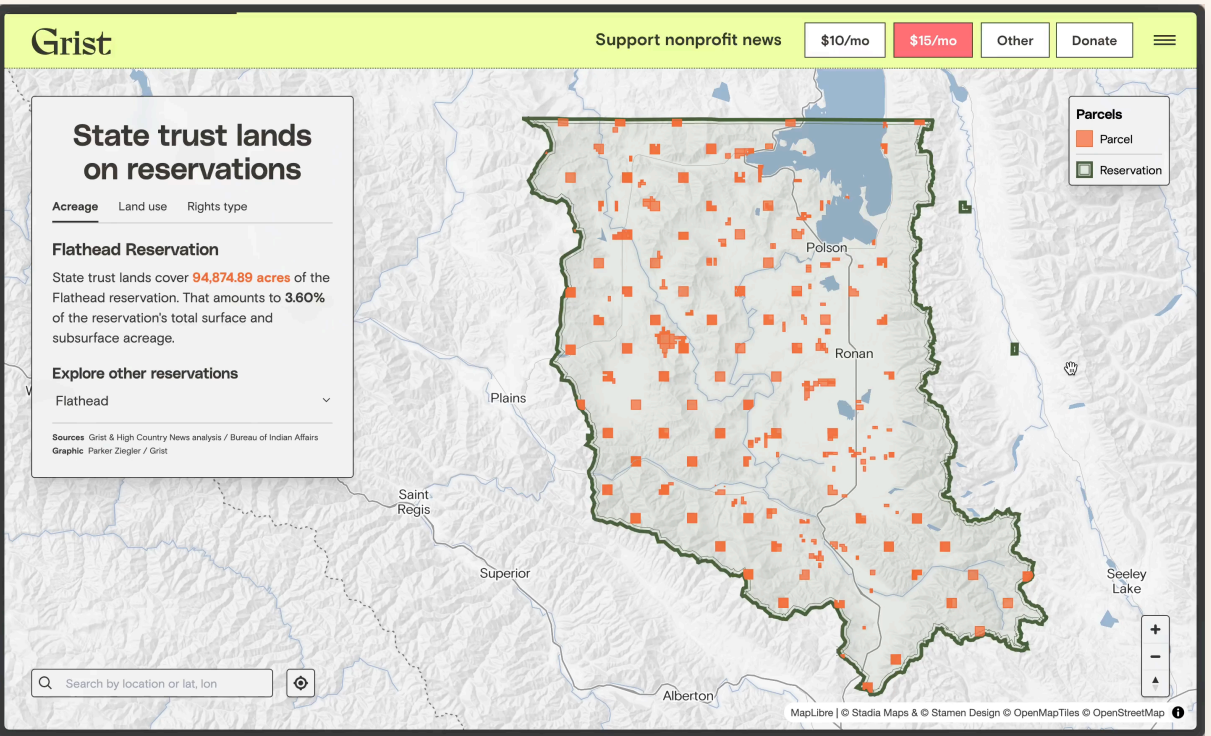
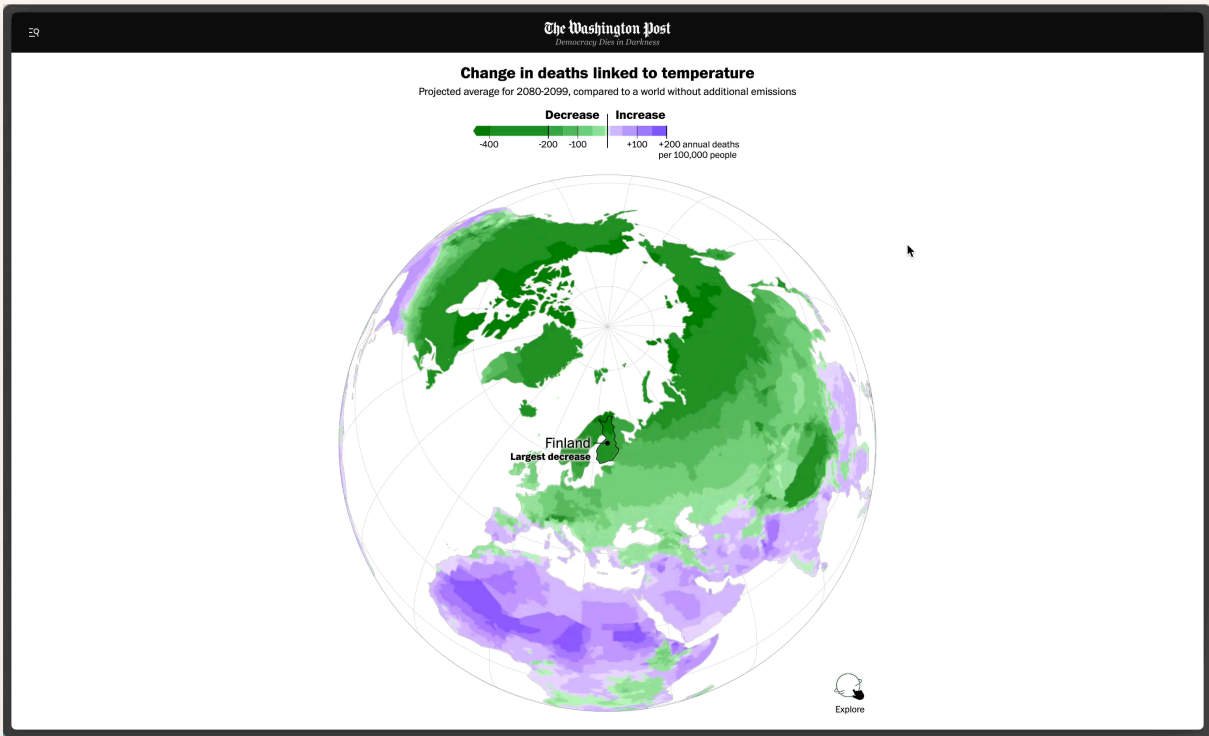




22 FEB 2022

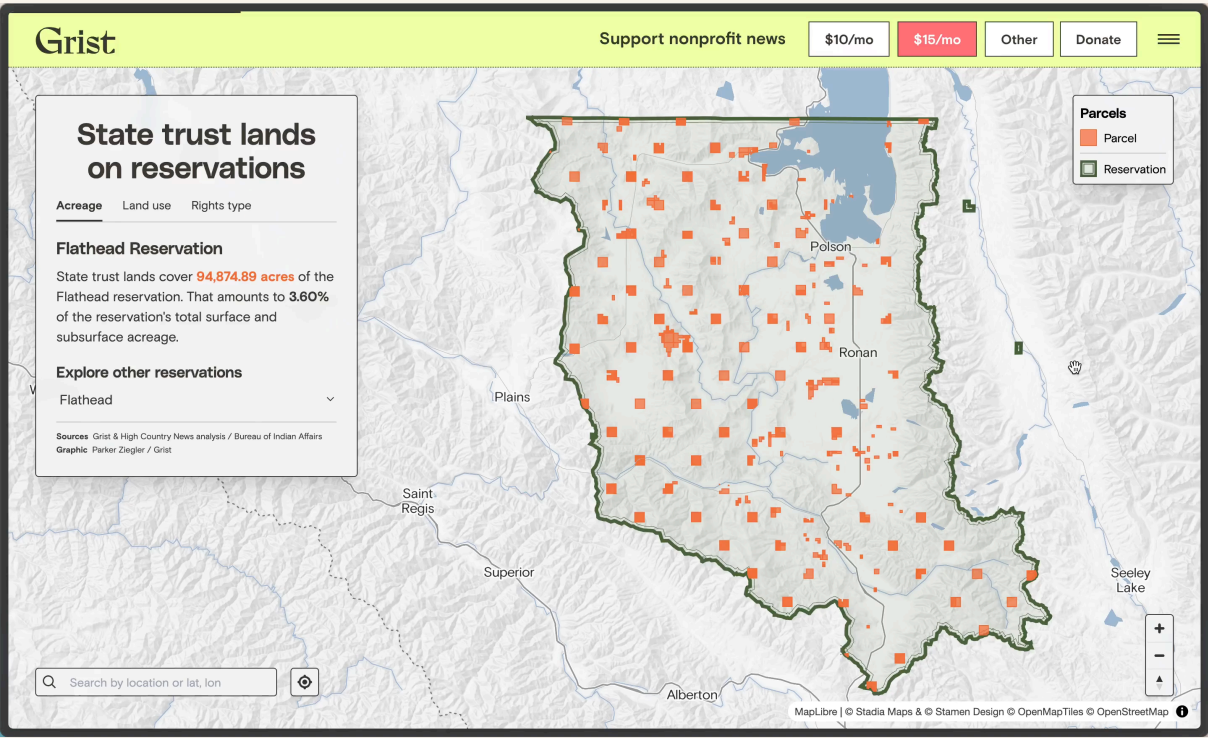
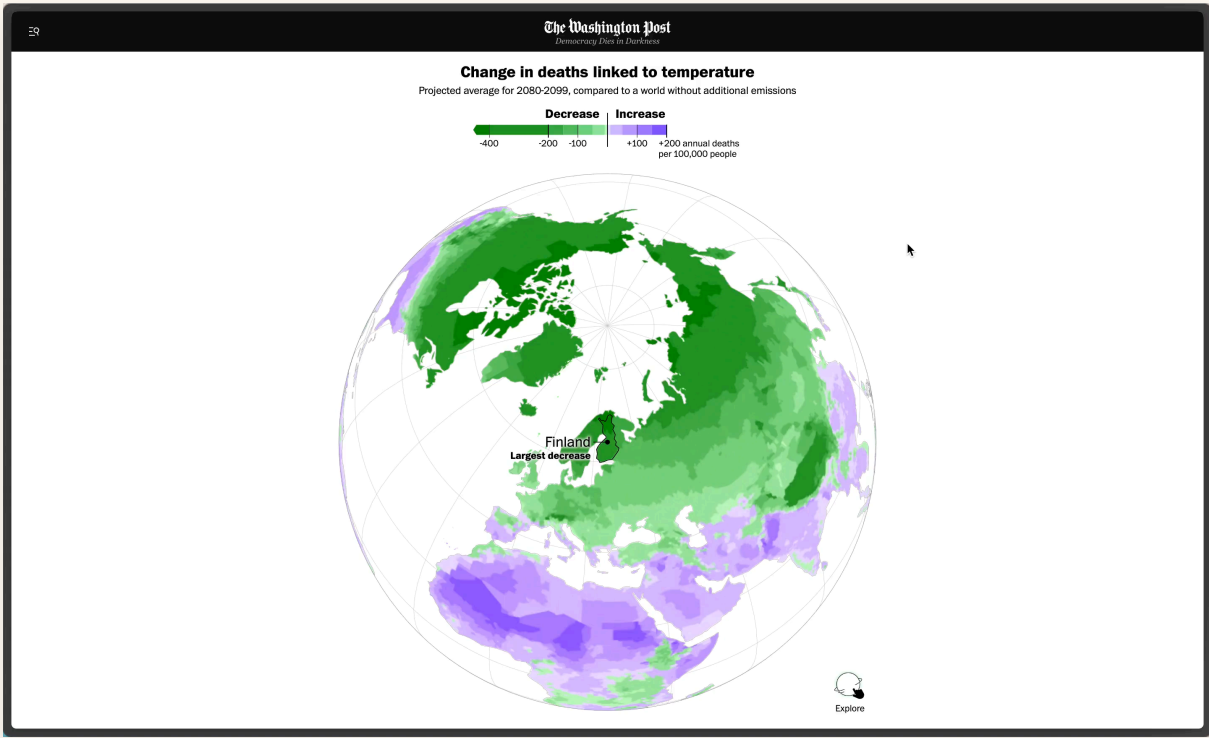


Beginning of my work with data journalists

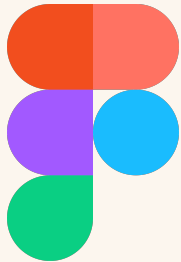


22 FEB 2022

Beginning of my work with data journalists



TOOLS FOR PROTOTYPING
Direct Manipulation GUIs

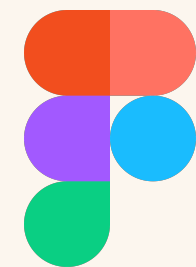


technically complex
expensive
time-consuming

Crossing this gap is **painful!**

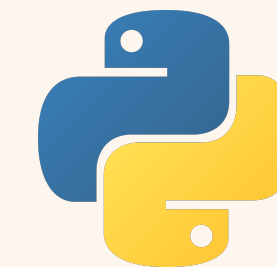
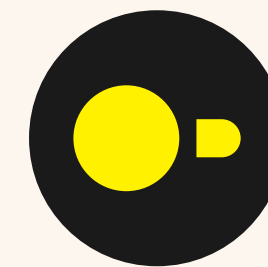
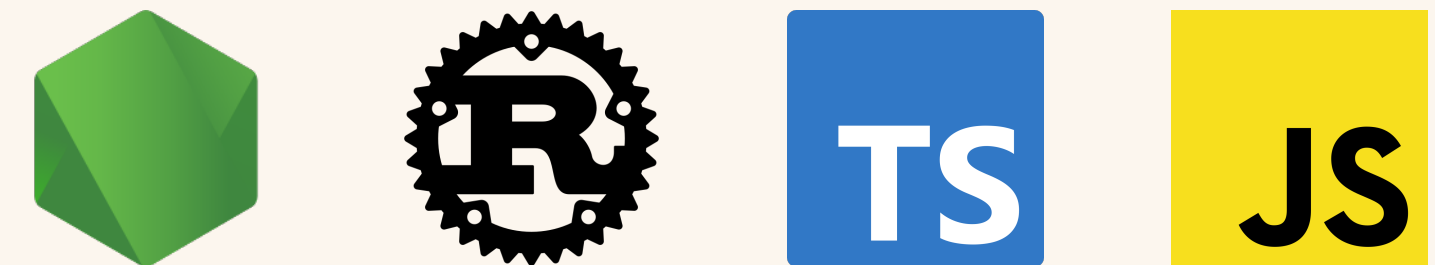
TOOLS FOR PROTOTYPING

Direct Manipulation GUIs



TOOLS FOR PRODUCTION

Programs



THE REAL QUESTION

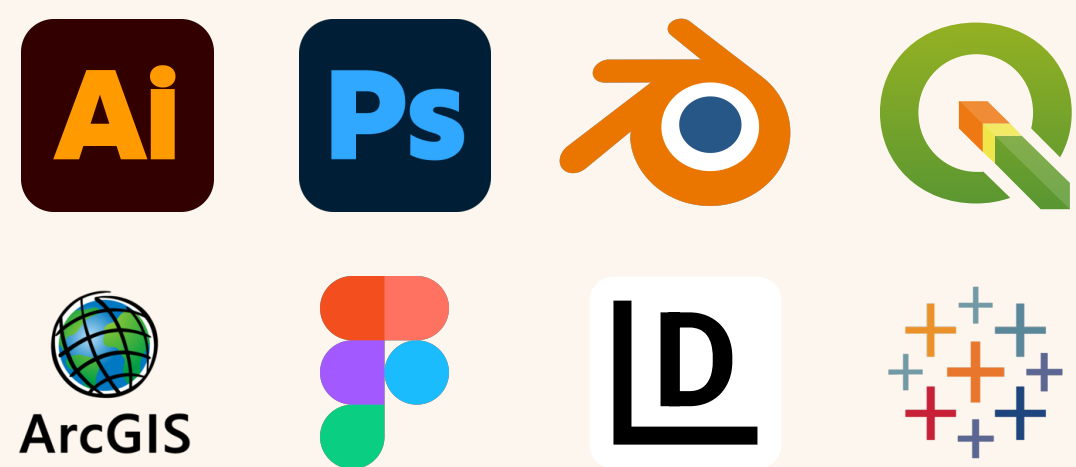
Can we develop programming systems that allow programmers to use familiar interactions from GUIs while automatically generating correct, readable, and performant code?

technically complex
expensive
time-consuming

Crossing this gap is painful!

TOOLS FOR PROTOTYPING

Direct Manipulation GUIs



TOOLS FOR PRODUCTION

Programs



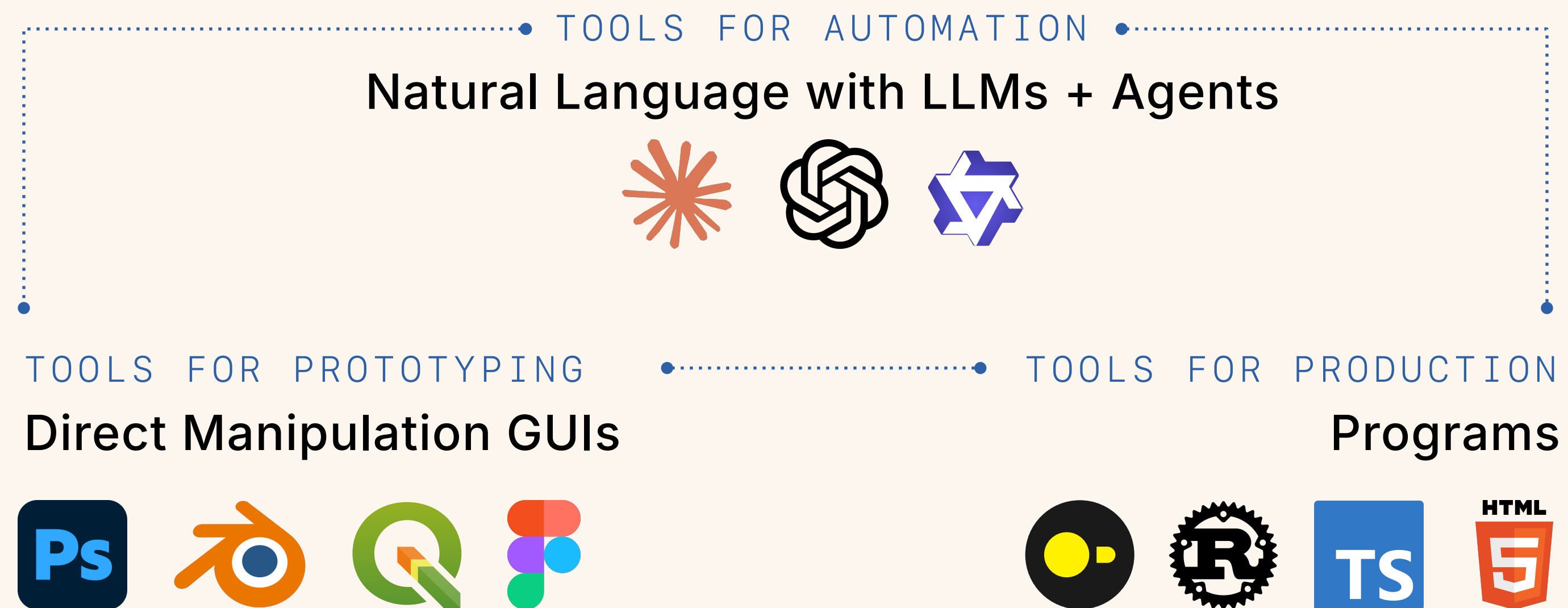
THE REAL QUESTION

Can we develop programming systems that allow programmers to use familiar interactions from GUIs while automatically generating correct, readable, and performant code?



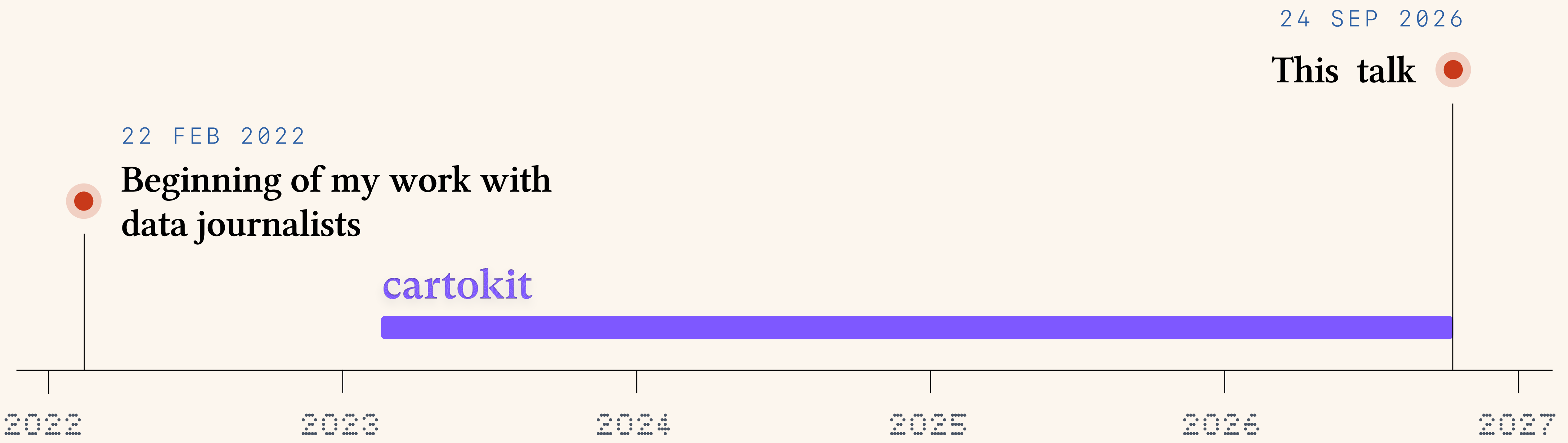
THE REAL, REAL QUESTION

Can we develop programming systems that allow programmers to use **multiple different modalities** while automatically generating correct, readable, and performant code?



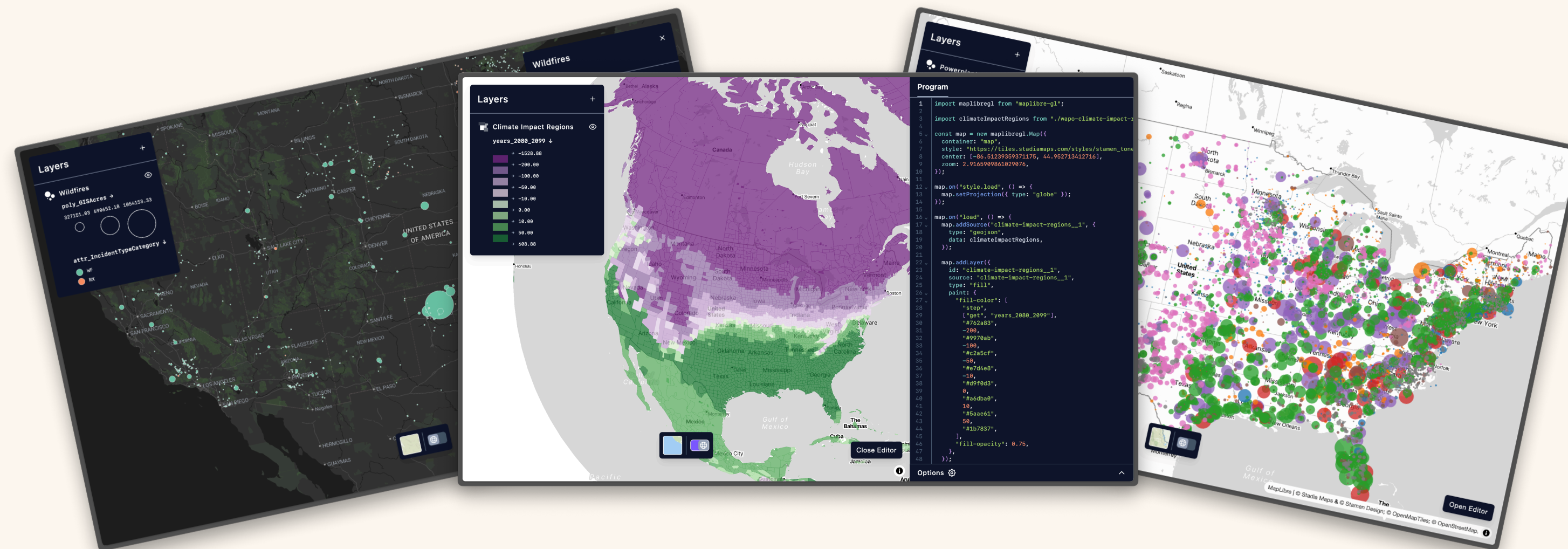
THE REAL, REAL QUESTION

Can we develop programming systems that allow programmers to use multiple different modalities while automatically generating correct, readable, and performant code?



cartokit

Multimodal programming for interactive cartography on the web



The image shows four 5x5 dot patterns arranged horizontally. Each pattern is composed of dark blue dots on a light beige background. The first pattern represents the digit '2', the second represents '0', the third represents '2', and the fourth represents '2'.

A 4x4 grid of dots forming the numbers 2, 0, 2, and 4.

A 4x4 grid of dots forming the number 2025. The first column has 4 dots, the second has 4 dots, the third has 4 dots, and the fourth has 4 dots.



A 4x4 grid of dots forming the number 2027. The first digit '2' is formed by dots at (1,1), (1,2), (1,3), (2,1), (2,2), (2,3), (3,1), (3,2), (3,3), (4,1), (4,2), (4,3). The second digit '0' is formed by dots at (1,4), (1,5), (1,6), (2,4), (2,5), (2,6), (3,4), (3,5), (3,6), (4,4), (4,5), (4,6). The third digit '2' is formed by dots at (1,7), (1,8), (1,9), (2,7), (2,8), (2,9), (3,7), (3,8), (3,9), (4,7), (4,8), (4,9). The fourth digit '7' is formed by dots at (1,10), (1,11), (1,12), (1,13), (1,14), (2,10), (2,11), (2,12), (2,13), (2,14), (3,10), (3,11), (3,12), (3,13), (3,14), (4,10), (4,11), (4,12), (4,13), (4,14).

Demo

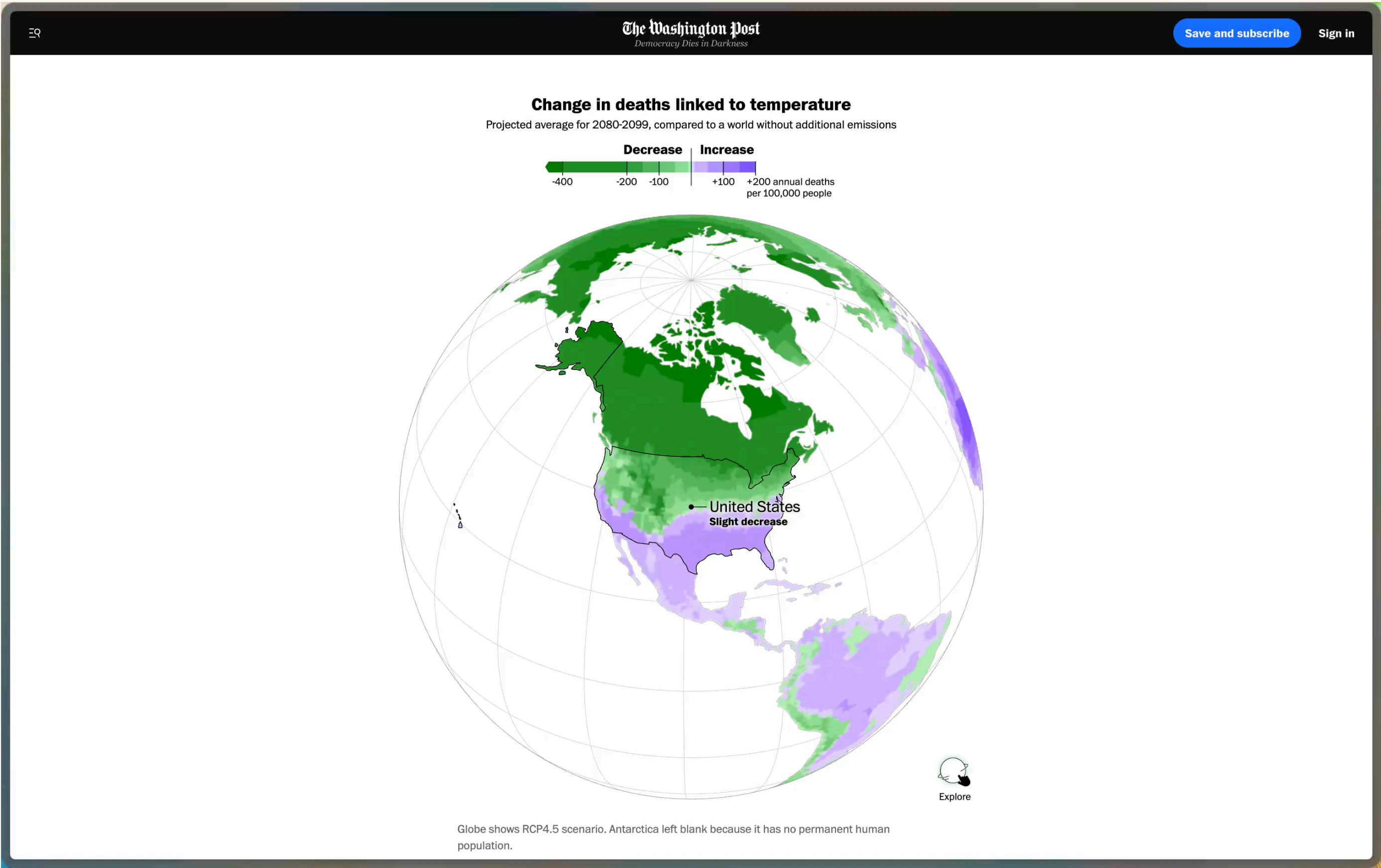


Will global warming make
temperature less deadly?

Harry Stevens

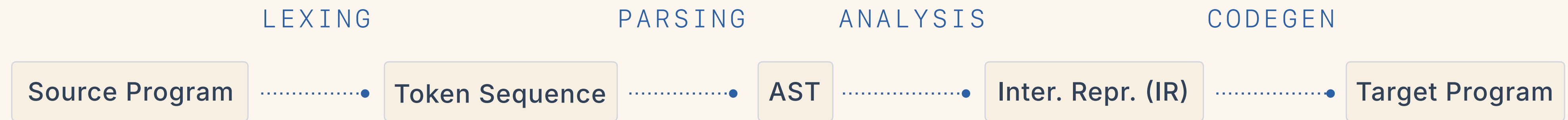
COOL, BUT...

How does it work?



COOL, BUT...

How does it work?



- A “traditional” compiler for text

COOL, BUT...

How does it work?

MULTIPLE MODALITIES

Natural Language

Direct Manipulation

LEXING

PARSING

ANALYSIS

CODEGEN

Source Program

Token Sequence

AST

Inter. Repr. (IR)

Target Program

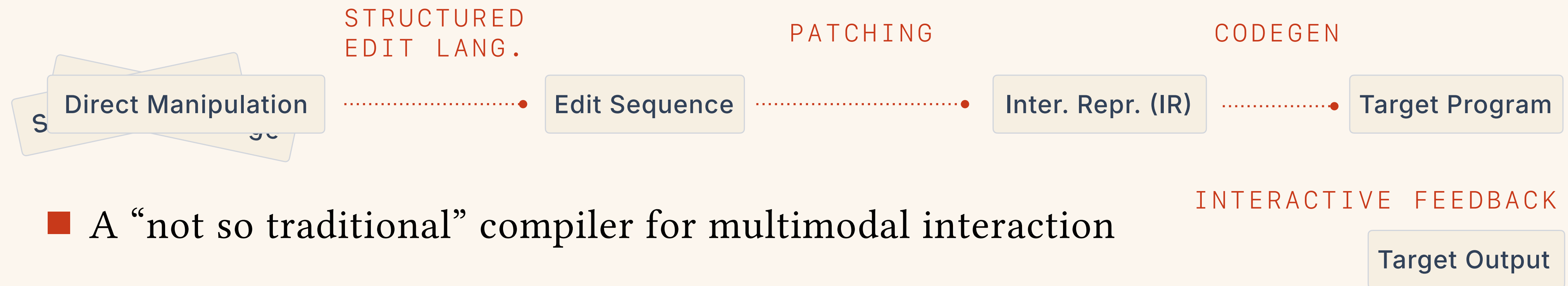
- A “traditional” compiler for text

INTERACTIVE FEEDBACK

Target Output

COOL, BUT...

How does it work?



- A “not so traditional” compiler for multimodal interaction

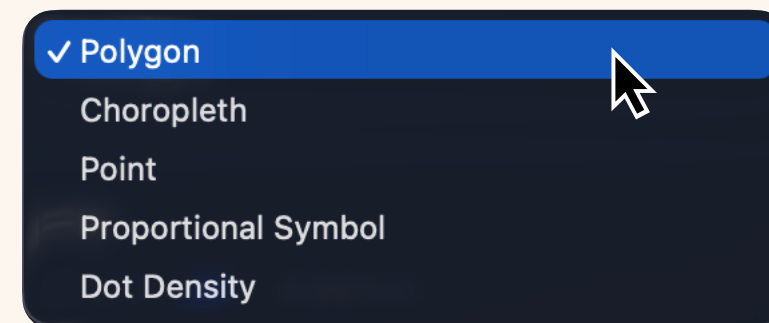
COOL, BUT...

How does it work?

KEY IDEA

Edits comprise structured data, not text diffs.

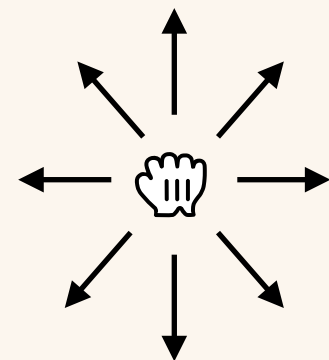
Set Layer Type



Set Projection



Pan Map



```
{
  type: "layer-type",
  layerId: "climate__1",
  payload: {
    sourceLayerType: "Polygon",
    targetLayerType: "Choropleth"
  }
}
```

```
{
  type: "projection",
  payload: {
    projection: "globe"
  }
}
```

```
{
  type: "center",
  payload: {
    center: [-118.37, 39.52]
  }
}
```

STRUCTURED
EDIT LANG.

Direct Manipulation

Edit Sequence

COOL, BUT...

How does it work?

```
{  
  type: "layer-type",  
  layerId: "climate__1",  
  payload: {  
    sourceLayerType: "Polygon",  
    targetLayerType: "Choropleth"  
  }  
}
```

```
{  
  type: "projection",  
  payload: {  
    projection: "globe"  
  }  
}
```

```
{  
  type: "center",  
  payload: {  
    center: [-118.37, 39.52]  
  }  
}
```

patch

patch

patch

IR

IR'

IR''

IR'''

PATCHING

Direct Manipulation

Edit Sequence

Inter. Repr. (IR)

COOL, BUT...

How does it work?

```
{
  type: "layer-type",
  layerId: "climate__1",
  payload: {
    sourceLayerType: "Polygon",
    targetLayerType: "Choropleth"
  }
}
```

Edit Sequence

patch

```
{
  center: [-106.1086, 37.7531],
  zoom: 4,
  layers: [
    {
      type: "Polygon",
      style: {
        fill: {
          color: "#8638e5",
          opacity: 0.75,
        },
      },
    },
    ...
  ],
}
```

IR

```
{
  center: [-106.1086, 37.7531],
  zoom: 4,
  layers: [
    {
      type: "Choropleth",
      style: {
        fill: {
          attribute: "years_2020_2039",
          scale: "Quantile",
          count: 8,
          scheme: "schemePuGr"
        },
      },
    },
    ...
  ],
}
```

IR'

Inter. Repr. (IR)

COOL, BUT...

How does it work?

```
{
  type: "layer-type",
  layerId: "climate__1",
  payload: {
    sourceLayerType: "Polygon",
    targetLayerType: "Choropleth"
  }
}
```

```
{
  type: "projection",
  payload: {
    projection: "globe"
  }
}
```

```
{
  type: "center",
  payload: {
    center: [-118.37, 39.52]
  }
}
```

Direct Manipulation

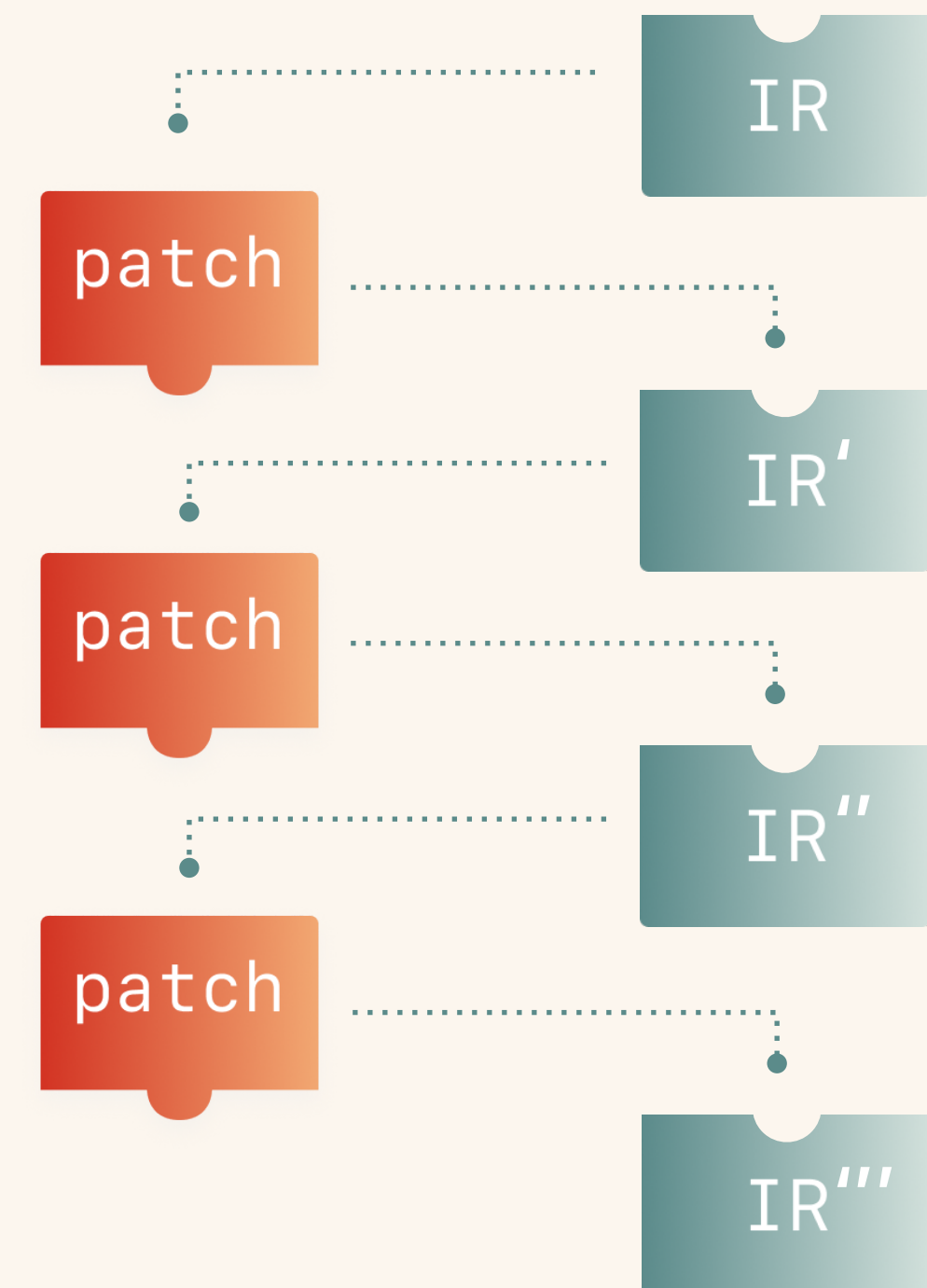
Edit Sequence

Inter. Repr. (IR)

PATCHING

KEY IDEA

patch defines a minimal IR transformation based on the edit.

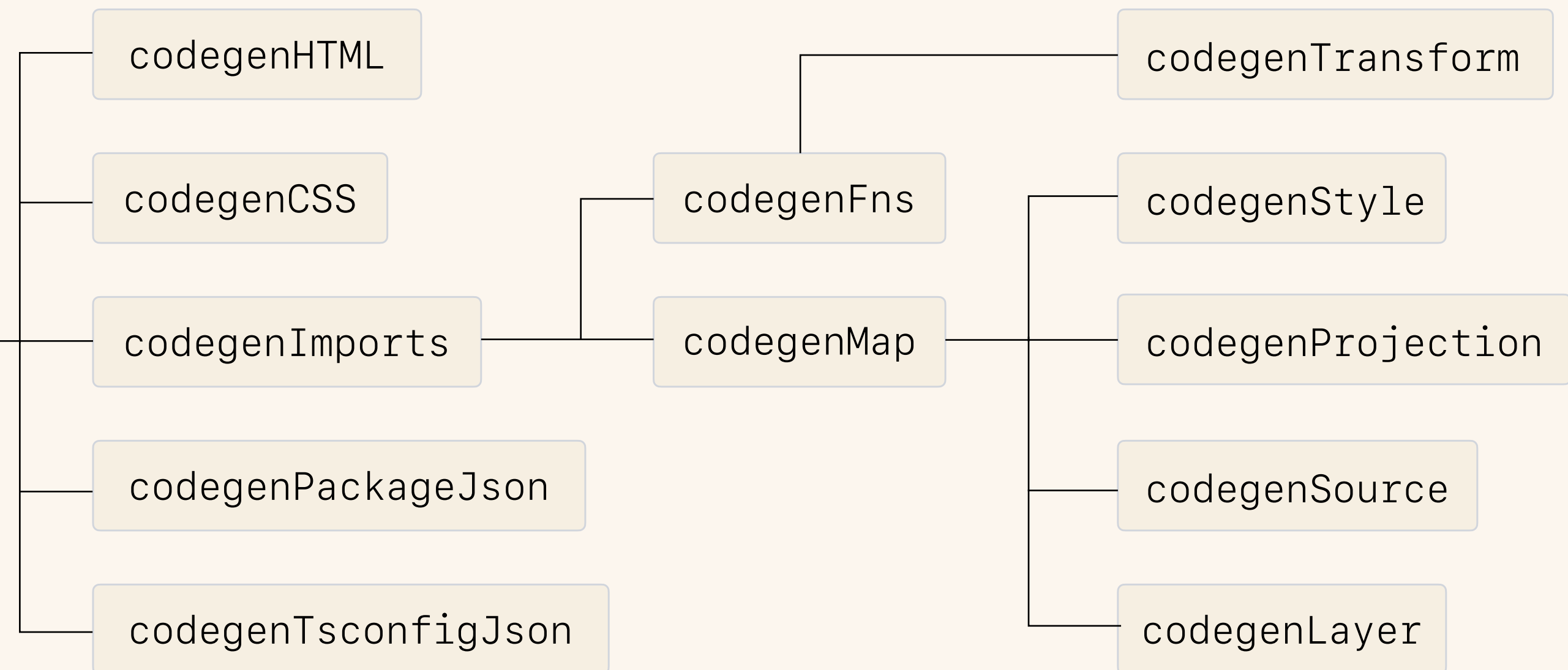


COOL, BUT...

How does it work?

```
{
  center: [-118.37, 39.52],
  zoom: 4,
  projection: "globe",
  "",
  layers: [
    {
      type: "Choropleth",
      style: {
        fill: {
          attribute: "years_2020_2039",
          scale: "Quantile",
          count: 8,
          scheme: "schemePuGr"
        },
        ""
      },
      },
    ""
  ],
  ""
]
```

IR'''



CODEGEN

Direct Manipulation

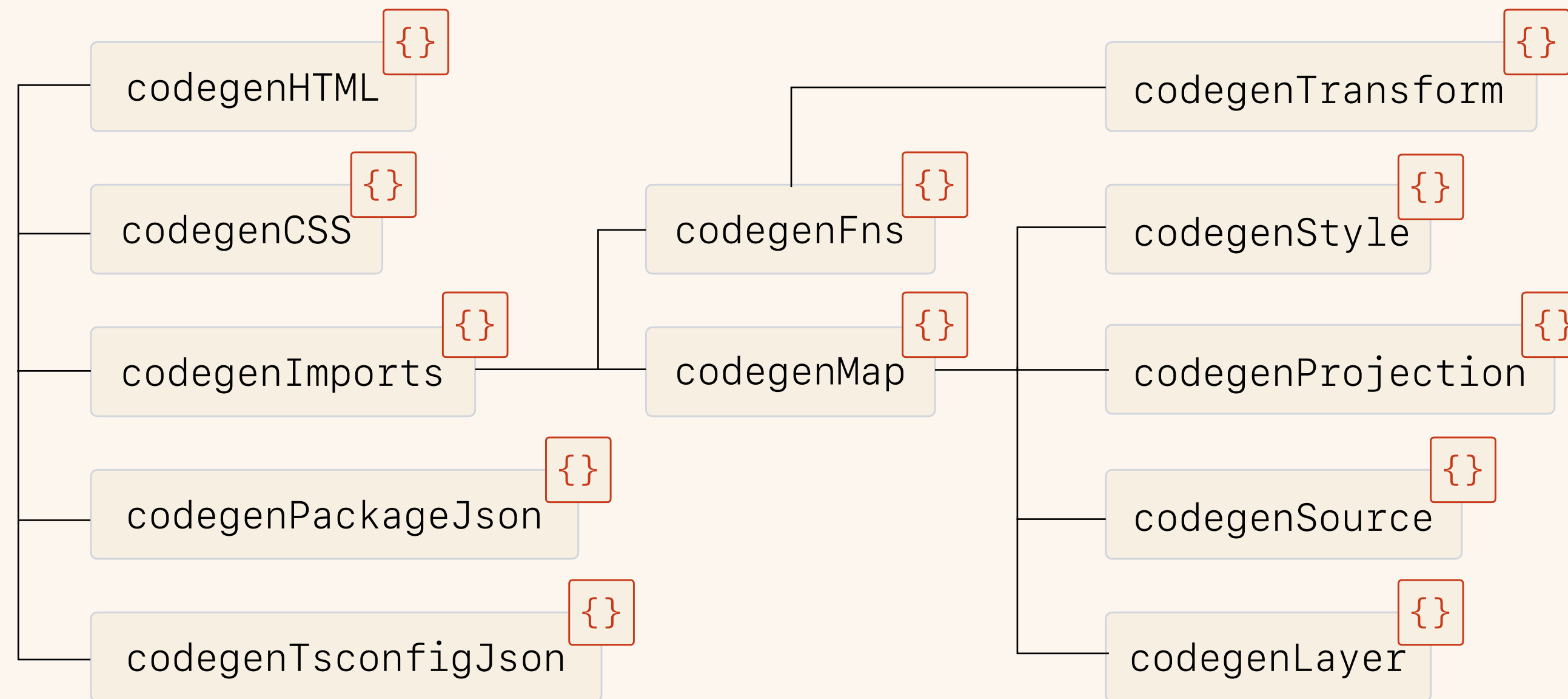
Edit Sequence

Inter. Repr. (IR)

Target Program

COOL, BUT...

How does it work?



CODEGEN

Direct Manipulation

Edit Sequence

Inter. Repr. (IR)

Target Program

COOL, BUT...

How does it work?

KEY IDEA

Callees generate **program text**, callers define **program location**.

```
import maplibregl from "maplibre-gl";

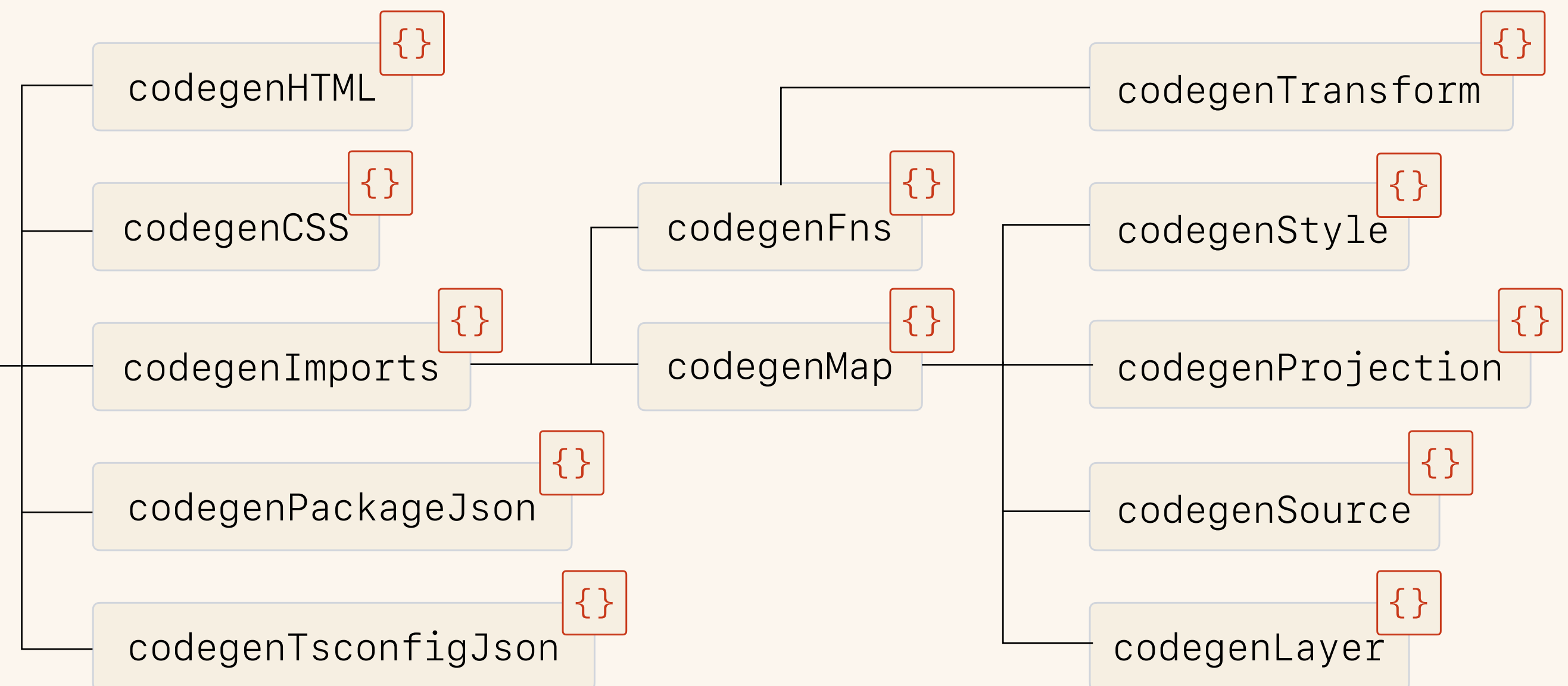
import "maplibre-gl/dist/maplibre-gl.css";
import "./style.css";

const map = new maplibregl.Map({
  container: "map",
  style: "https://tiles.stadiamaps.com/styles/stamen_...",
  center: [-95.02492187500002, 39.5],
  zoom: 4,
});

map.on("style.load", () => {
  map.setProjection({ type: "globe" });
});

map.on("load", () => {
  map.addSource("climate-impact-regions__1", {
    type: "geojson",
    data: "https://pub-71829.r2.dev/wapo-climate-imp...",
  });

  map.addLayer({
    id: "climate-impact-regions__1",
    source: "climate-impact-regions__1",
    type: "fill",
    paint: {
      "fill-color": [
        "step",
        ["get", "years_2080_2099"],
        "#1b7837",
        -400,
      ],
    },
  });
});
```



CODEGEN

Direct Manipulation

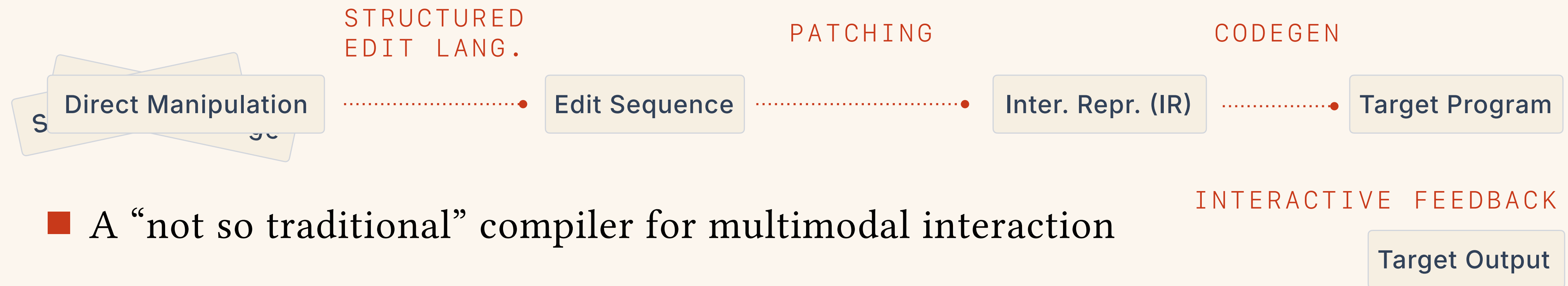
Edit Sequence

Inter. Repr. (IR)

Target Program

COOL, BUT...

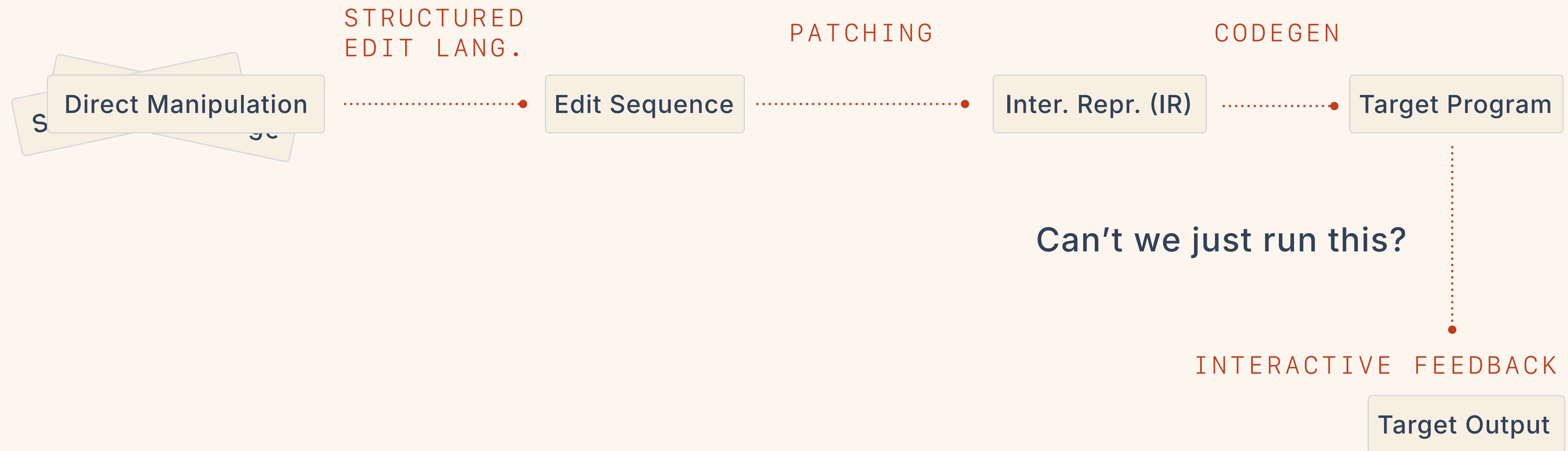
How does it work?



■ A “not so traditional” compiler for multimodal interaction

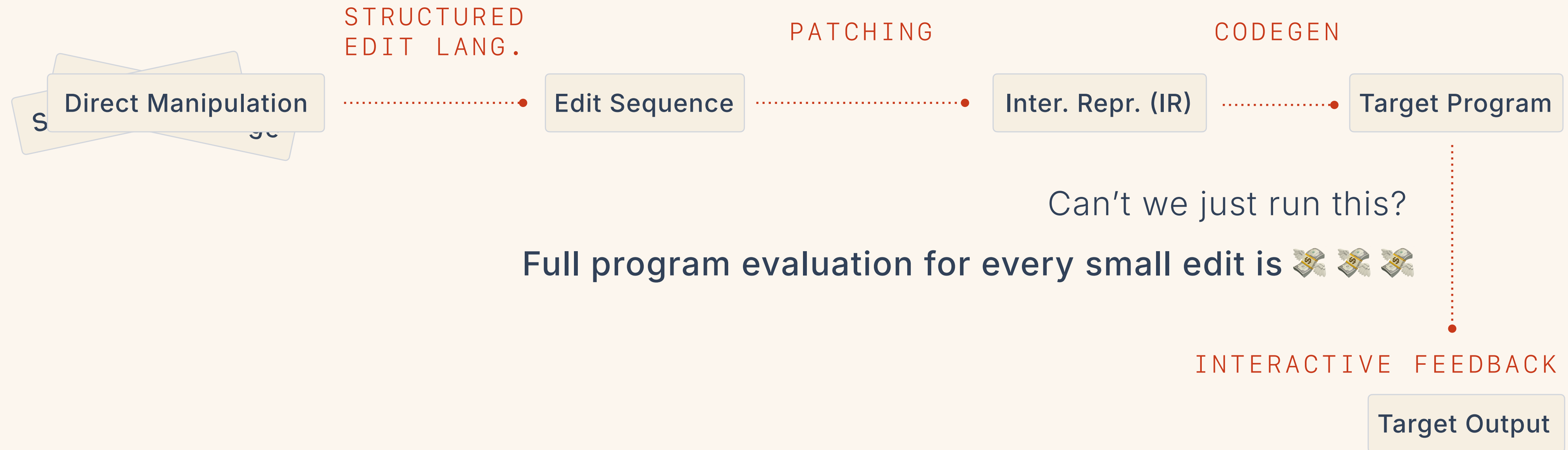
COOL, BUT...

How does it work?



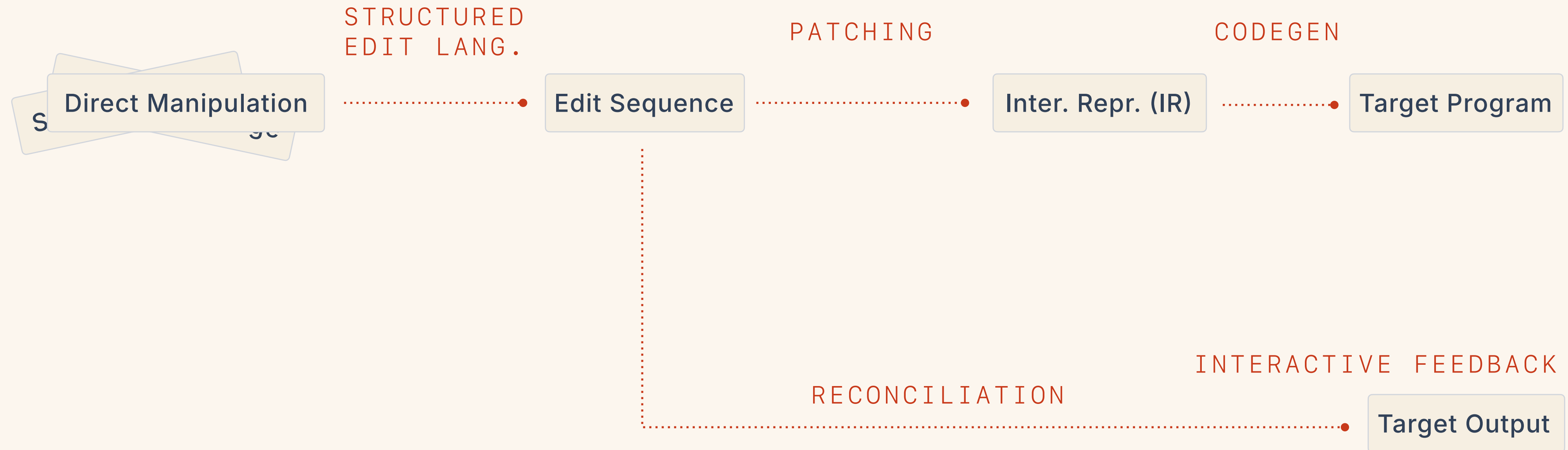
COOL, BUT...

How does it work?



COOL, BUT...

How does it work?



COOL, BUT...

How does it work?

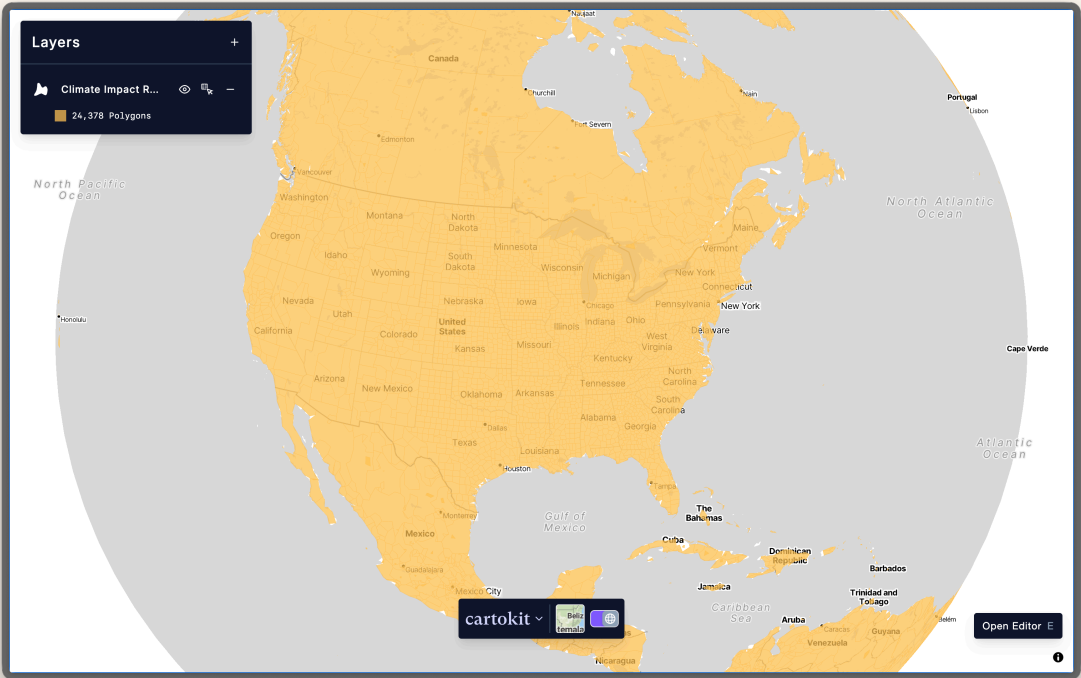
```
{
  type: "layer-type",
  layerId: "climate__1",
  payload: {
    sourceLayerType: "Polygon",
    targetLayerType: "Choropleth"
  }
}
```

Edit Sequence

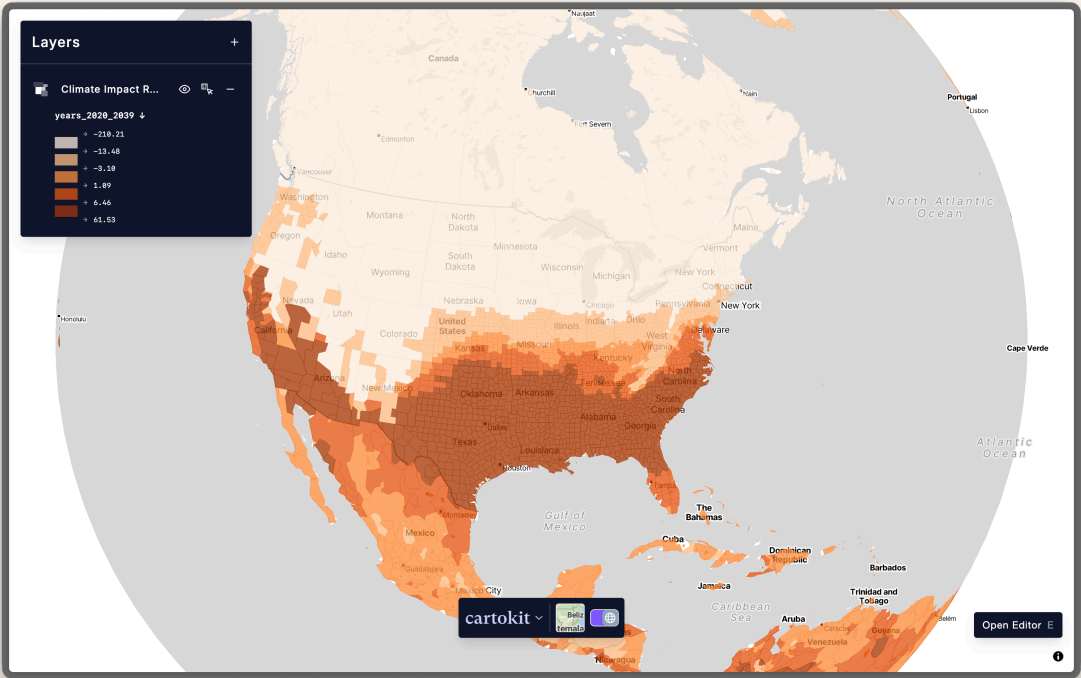


Update Polygon fill-color

```
map.value?.setPaintProperty(
  targetLayer.id,
  'fill-color',
  deriveColorScale(targetLayer)
);
```



Target Output



Target Output'

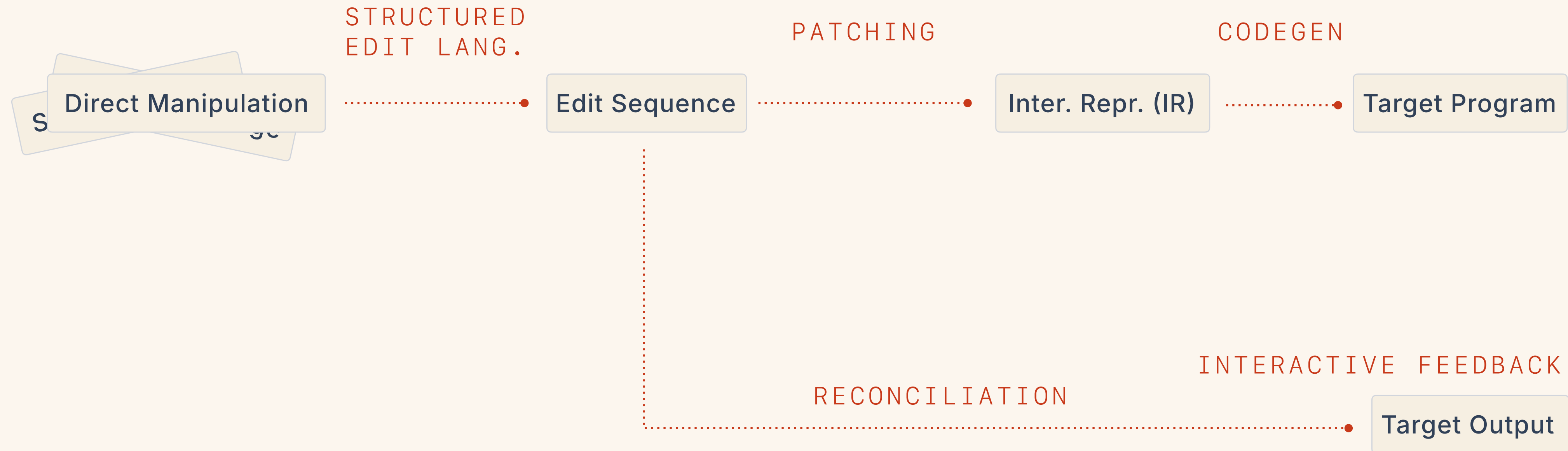
COOL, BUT...

How does it work?

KEY IDEA

recon defines a minimal output update based on the edit.

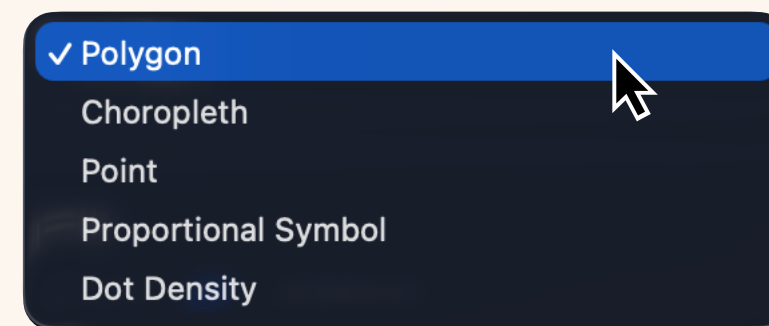
VDOM reconciliation, but the diffs are already defined!



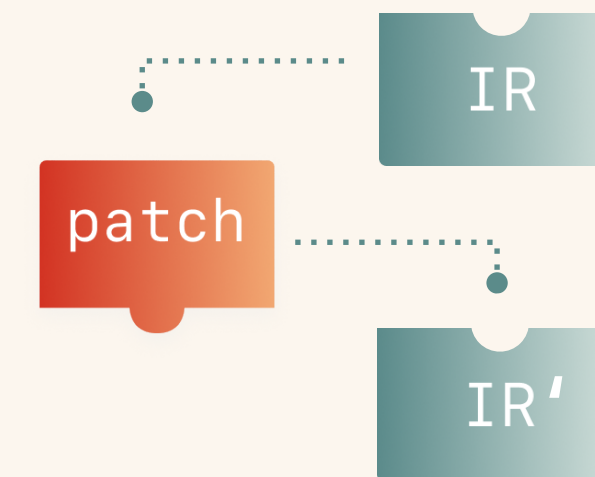
COOL, BUT...

How does it work?

Set Layer Type



```
{  
  type: "layer-type",  
  layerId: "climate__1",  
  payload: {  
    sourceLayerType: "Polygon",  
    targetLayerType: "Choropleth"  
  }  
}
```



STRUCTURED
EDIT LANG.

PATCHING

CODEGEN

Direct Manipulation

Edit Sequence

Inter. Repr. (IR)

Target Program

■ A “not so traditional” compiler for multimodal interaction

INTERACTIVE FEEDBACK

Target Output

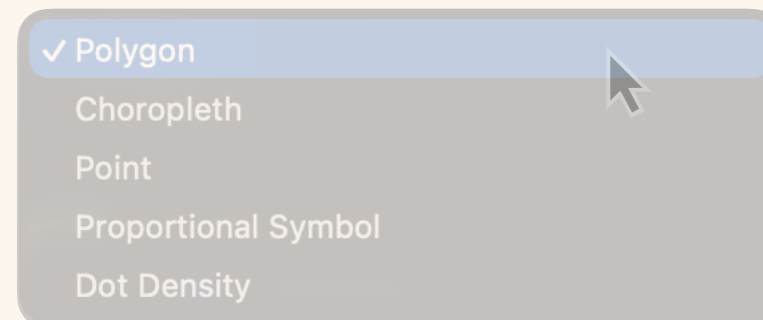
RECONCILIATION



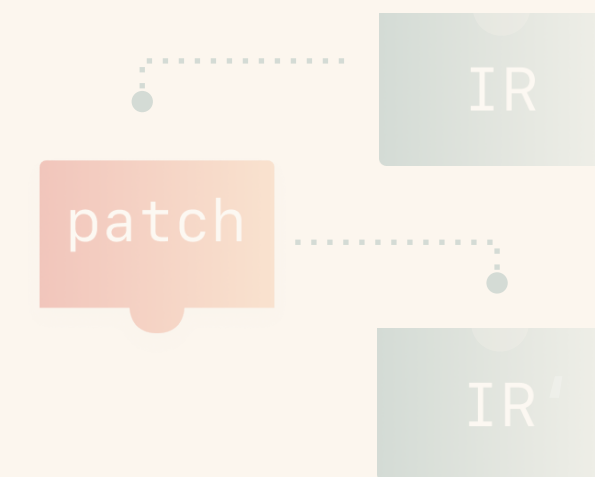
COOL, BUT...

How does it work?

Set Layer Type



```
{  
  type: "layer-type",  
  layerId: "climate__1",  
  payload: {  
    sourceLayerType: "Polygon",  
    targetLayerType: "Choropleth"  
  }  
}
```



STRUCTURED
EDIT LANG.

PATCHING

CODEGEN

Direct Manipulation

Edit Sequence

Inter. Repr. (IR)

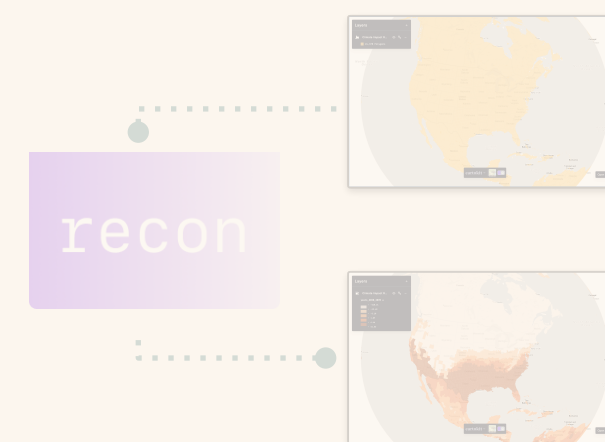
Target Program

■ A “not so traditional” compiler for multimodal interaction

INTERACTIVE FEEDBACK

Target Output

RECONCILIATION



COOL, BUT...

How does it work?

Prompt

Transition the climate layer to a choropleth map, coloring by the years_2080_2099 property.



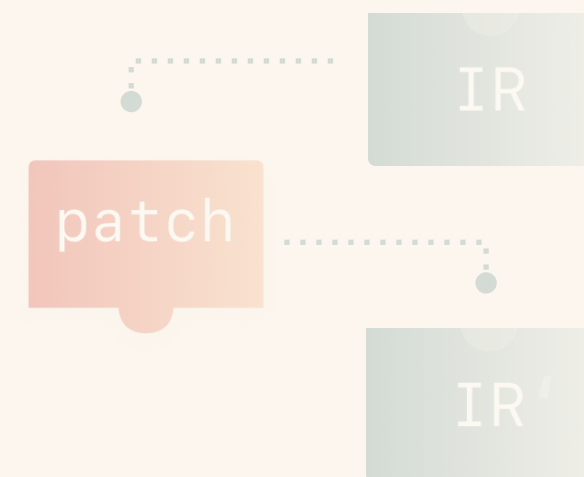
Natural Language

```
{
  type: "layer-type",
  layerId: "climate__1",
  payload: {
    sourceLayerType: "Polygon",
    targetLayerType: "Choropleth"
  }
}
```

CONSTRAINED
DECODING

Edit Sequence

PATCHING



CODEGEN

Inter. Repr. (IR)

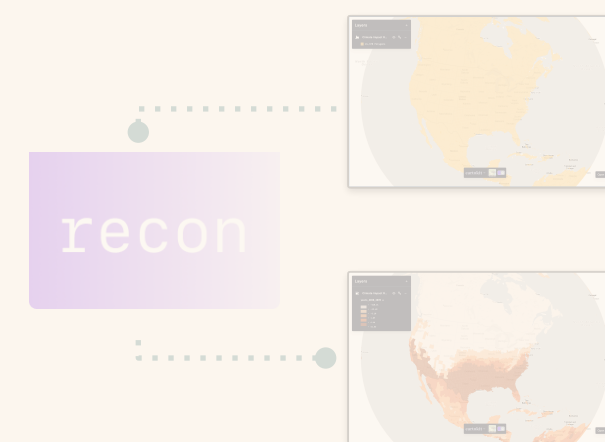
Target Program

■ A “not so traditional” compiler for multimodal interaction

INTERACTIVE FEEDBACK

Target Output

RECONCILIATION



COOL, BUT...

How does it work?

Prompt

Transition the climate layer to a choropleth map, coloring by the years_2080_2099 property.



System Prompt

...

Diffs

A diff is a single, atomic, structured edit. Emit diffs in the order they should be applied. Later diffs may have a dependency on previous diffs, (e.g., "...")

CFG Annotated with Semantic Constraints

```
const LayerTypeSchema = z.enum([
  "Choropleth",
  "Dot Density",
  "Heatmap",
  "Line",
  "Point",
  "Polygon",
  "Proportional Symbol"
]);

const LayerTypeDiff = z.object({
  type: z.literal("layer-type"),
  layerId: LayerIdSchema,
  payload: z.object({
    sourceLayerType: LayerTypeSchema,
    targetLayerType: LayerTypeSchema
  })
});

// +64 more diffs!
```

```
{
  type: "projection",
  payload: {
    projection: "globe"
  }
}
```

```
{
  type: "center",
  payload: {
    center: [-118.37, 39.52]
  }
}
```

```
{
  type: "layer-type",
  layerId: "climate__1",
  payload: {
    sourceLayerType: "Polygon",
    targetLayerType: "Choropleth"
  }
}
```

Model

CONSTRAINED
DECODING

Natural Language

Edit Sequence

COOL, BUT...

How does it work?

Prompt

Transition the climate layer to a choropleth map, coloring by the years_2080_2099 property.



System Prompt

...

Diffs

A diff is a single, atomic, structured edit. Emit diffs in the order they should be applied. Later diffs may have a dependency on previous diffs, (e.g., "...")

Natural Language

CFG Annotated with Semantic Constraints

```
const LayerTypeSchema = z.enum([
  "Choropleth",
  "Dot Density",
  "Heatmap",
  "Line",
  "Point",
  "Polygon",
  "Proportional Symbol"
]);

const LayerTypeDiff = z.object({
  type: z.literal("layer-type"),
  layerId: LayerIdSchema,
  payload: z.object({
    sourceLayerType: LayerTypeSchema,
    targetLayerType: LayerTypeSchema
  })
});

// +64 more diffs!
```

Model

Choose diffs!

This sounds a lot like Jev and System One models!

Edit Sequence

```
{
  type: "projection",
  payload: {
    projection: "globe"
  }
}
```

```
{
  type: "center",
  payload: {
    center: [-118.37, 39.52]
  }
}
```

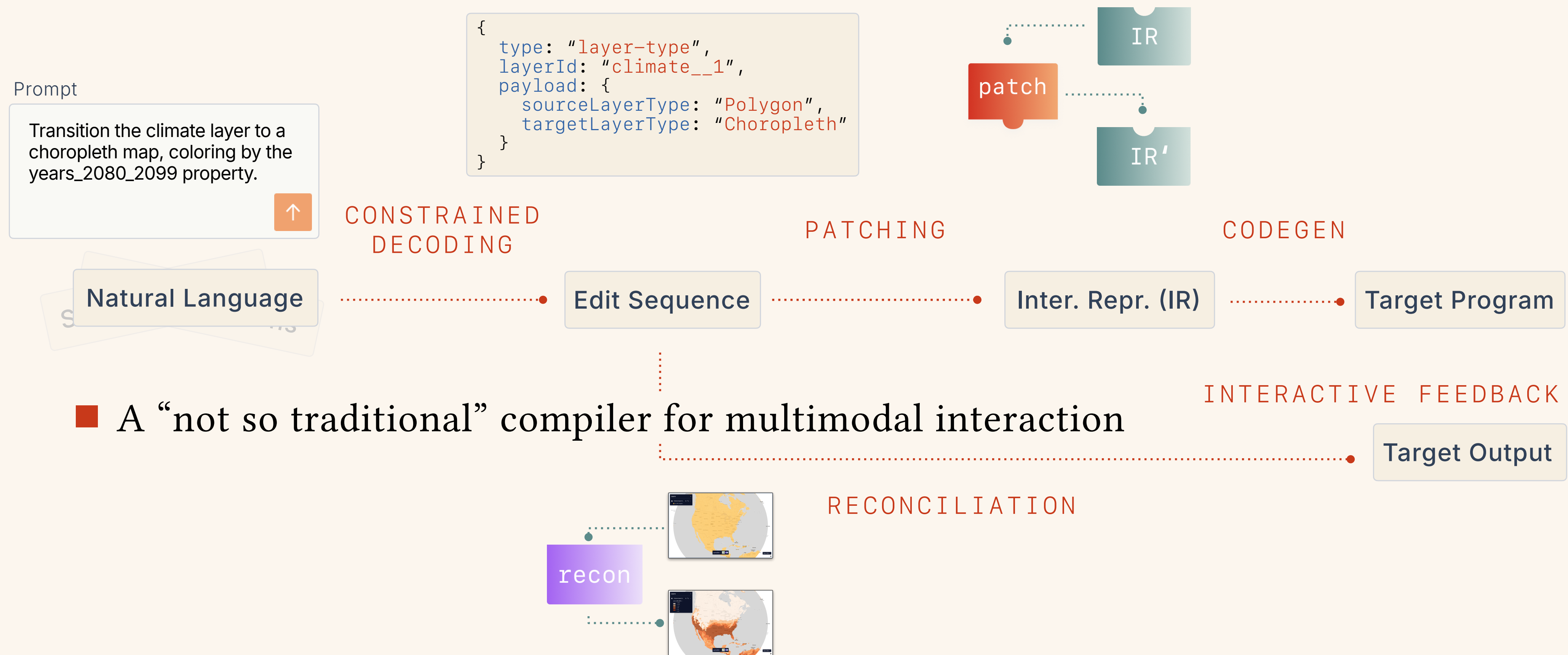
```
{
  type: "layer-type",
  layerId: "climate__1",
  payload: {
    sourceLayerType: "Polygon",
    targetLayerType: "Choropleth"
  }
}
```


COOL, BUT...

How does it work?

KEY IDEA

Use a constrained **edit language** as the surface area for an LLM.



COOL, BUT...

How does it work?

THE TAKEAWAY

Use a constrained **edit language** as the surface area for an LLM.

all modalities

MULTIPLE MODALITIES

Natural Language

• CONSTRAINED DECODING

Direct Manipulation

• STRUCTURED
EDIT LANG.

TEXT EDITING

PATCHING

CODEGEN

Source Program

Edit Sequence

Inter. Repr. (IR)

Target Program

■ A “not so traditional” compiler for multimodal interaction

INTERACTIVE FEEDBACK

Target Output

RECONCILIATION

THE REAL, REAL QUESTION

Can we develop programming systems that allow programmers to use **multiple different modalities** while automatically generating correct, readable, and performant code?

THE ANSWER

Absolutely! 

cartokit



2022

2023

2024

2025

2026

2027

Used by
journalists at



cartokit

<https://alpha.cartokit.dev/>
<https://github.com/parkerziegler/cartokit/>



Parker Ziegler
<https://parkie-doo.computer/>
peziegler@cs.berkeley.edu

