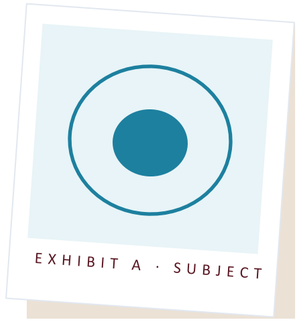
IOWA TECHNOLOGY & GEOSPATIAL CONFERENCE · THE MEADOWS, ALTOONA

From Plain English to GIS App in 30 minutes or less.



A live build with Claude Code, GeoBlazor & the ArcGIS Maps SDK.

FILED BY
dymaptic

PRESENTERS
Tim Purdum & Holly Kluever

COORDINATES
41.5868° N · 93.6250° W

ENGAGEMENT
ITAG · 2026 · LIVE · 30 MIN

SPEAKERS · 02 / 24 · THE INVESTIGATORS

Hello, we're dymaptic.
We love to build GIS software.

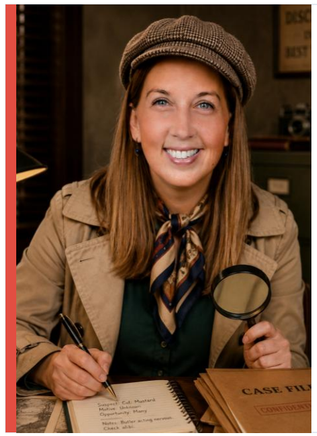
CO-PRESENTER · 01

Tim Purdum

Director of Product Development, dymaptic

Inventor of GeoBlazor

Designs custom GIS apps for clients



CO-PRESENTER · 02

Holly Kluever

Chief Operating Officer, dymaptic

Manages business ops, marketing, HR

GIS enthusiast

THE CASE FILE · ~24,000 IOWA BRIDGES

What we're building, live.

Iowa's bridges, scored on a map. Today's dataset: Iowa DOT bridge inspections — public record, ~24,000 bridges, scored 0–9.

01 WebMap centered on Iowa, basemap toggle.

02 FeatureLayer of bridges, color-coded by condition rating.

03 Sidebar filters: condition · year built · county.

04 Popups: name, year, traffic, load rating, last inspection.

05 "Find Bridges at Risk" — SQL query, ring overlay.

→ FIND BRIDGES AT RISK

THE STAKEOUT · 04 / 24

The sentence is the seed.

EXHIBIT · PROMPT

> *Build me a Blazor map of Iowa's bridges from the National Bridge Inventory. Color them by condition — green for good, amber for fair, red for poor. Add a sidebar with filters for condition, year built, and county, and a button to find the bridges most at risk.*

This one sentence is everything I hand the agent to start. While Claude builds from it, I'll walk you through how it reads the prompt — describing the map, picking the data, and laying out the app before a single line is written by hand.

FIRST CLUE · WHAT I'M TYPING NOW

Watch us actually build it.

Switching to the terminal & the browser. If something breaks, you'll see how we recover.

12 MIN · LIVE

THE MOTIVE · 01 / 02

You know exactly *what you want* your GIS app to do.

THE MOTIVE · 02 / 02

You just don't always know how to tell a computer to do it.

THE USUAL SUSPECTS · 07 / 24

The translator was always the bottleneck.

From an idea on a Tuesday to a working mapping application was, optimistically, a couple of weeks.

STEP 01 · THE ASK

Analyst sketches what they want.

A whiteboard. A spec. A frantic Teams message. "Like Google Maps but for our bridges."

STEP 02 · THE TRANSLATE

Developer turns it into code.

Tickets. Sprints. Three back-and-forths over what "color-coded by condition" actually means.

STEP 03 · THE REVEAL

Analyst sees it.

"That's… not quite what I meant." Repeat steps 01–03. Sprint by sprint.

NEW LEADS · 08 / 24

Then, somewhere in the last 18 months …

The translator got really, really good.

THE TOOLKIT · 09 / 24

Three pieces. One workflow.

The stack on stage today.

CC

THE TRANSLATOR

Claude Code

An AI coding agent that lives in your terminal. Reads your files, writes your files, runs commands, asks questions when it's stuck.

GB

THE FRAMEWORK

GeoBlazor

Open source. The ArcGIS Maps SDK for JavaScript, wrapped for Blazor & .NET. Build serious GIS web apps in C#.

AG

THE ENGINE

ArcGIS Maps SDK

Esri's mapping SDK. WebMaps, layers, widgets, Arcade. The grown-up infrastructure underneath everything we build.

EXHIBIT A · 01 / 03

THE TRANSLATOR

Claude Code.

Not a chatbot. Not autocomplete. An agent that opens your project, edits files, runs your build, and tells you when it broke something.

MADE BY
ANTHROPIC

LIVES IN
YOUR TERMINAL

BEST FOR
MULTI-FILE EDITS

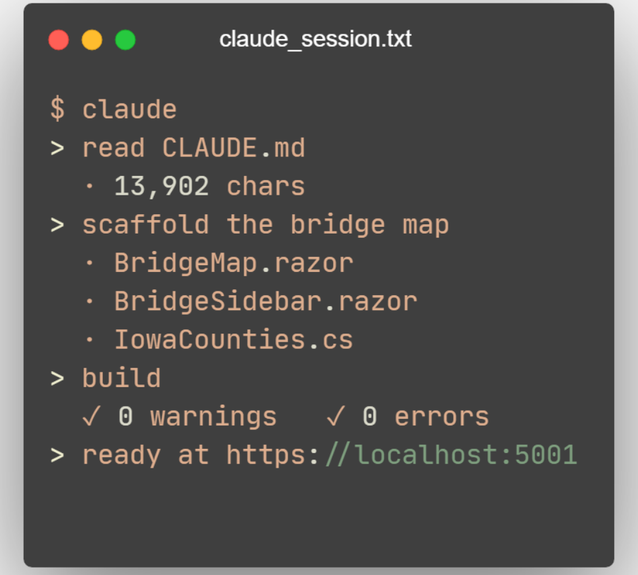


EXHIBIT B · 02 / 03

THE FRAMEWORK

GeoBlazor.

Open source. Built by us. Wraps the ArcGIS Maps SDK for JavaScript in Razor components so you can build serious GIS apps in C#.

LICENSE

MIT

REPOSITORY

GITHUB.COM/DYMAPTIC/GEOBLAZOR

BEST FOR

.NET TEAMS & ARCGIS SHOPS

home.razor

```
<MapView Longitude="-93.6"
         Latitude="42.0"
         Zoom="7">
  <ArcGISNavigationBasemap />
  <FeatureLayer Url="@Url">
    <UniqueValueRenderer
      ValueExpression="@Arcade"/>
  </FeatureLayer>
</MapView>
```

EXHIBIT C · 03 / 03

THE ENGINE

ArcGIS Maps SDK.

The grown-up underneath. WebMaps, FeatureLayers, widgets, popups — and Arcade, the expression language that paints every bridge by composite condition.

MADE BY
ESRI

FLAVORS
JS · .NET NATIVE

RENDERED BY
ARCADE EXPRESSION

arcade_condition.js

```
// Arcade · condition renderer
var d = IIF($feature.DECK_COND_058
        == 'N', 99,
        Number($feature.DECK_COND_058));
var c = Min(d, s, b);

When(c ⩽ 4, 'Poor',
     c ⩽ 6, 'Fair',
     c ⩽ 9, 'Good',
     'Unknown')
```

THE METHOD · 13 / 24

Three verbs. In order. On repeat.

The whole workflow.

VERB 01

Describe.

In plain English. Not pseudocode, not API names — just the map you can picture in your head.

VERB 02

Build.

Claude reads the docs, writes the components, wires the WebMap. You watch the diffs scroll by.

VERB 03

Refine.

"Smaller legend." "Add a county filter." "Highlight the scary bridges." One sentence at a time.

VERB 02 · BUILD

It writes. You watch.

build_log.txt

```
✦ planning · 6 steps identified
✓ scaffold BridgeMap.razor + Home.razor
✓ FeatureLayer + DefinitionExpression (Iowa only)
✓ UniqueValueRenderer keyed on Arcade condition
✓ BridgeSidebar: condition / year / county filters
✓ PopupTemplate: name, year, ADT, load rating, inspection
✓ "Find Bridges at Risk" — SQL query + ring overlay
… writing files…
✓ 6 files created · 2 files edited · build OK · 1.3 s
» "Map should be running at https://localhost:5001 — go look."
```

VERB 03 · REFINE

Talk to it like a junior teammate.

REFINEMENT 01

"The legend's too big."

Claude resizes it. Moves it. Asks if you'd rather collapse it into a toggle.

REFINEMENT 02

"Every bridge is grey."

You describe what you see. It finds the silent Arcade bug the compiler missed.

REFINEMENT 03

"Group the county filter alphabetically."

Sentence in, sorted dropdown out. Tells you what it changed so you can review the diff.

REFINEMENT 04

"That broke the popup."

It reads the error. It explains the fix. You move on.

CROSS-EXAMINATION · 18 / 24

It is not, in fact, magic.

When it goes wrong.

FAILURE MODE 01

Wrong API. Compiler catches.

Reaches for ClassBreaksRenderer. Not in GeoBlazor 4.4.4. Library version drift, not invention. Build errors within seconds; fixed on the next pass.

FAILURE MODE 02

Compiles. Runs. Every bridge is grey.

Arcade ValueExpression with a wrong field reference. Map loads. Legend shows four categories. Every bridge classified "Unknown." The dangerous failure.

FAILURE MODE 03

Popups open empty. Layer shows up grey.

FieldInfo declared as direct PopupTemplate children (no body). FeatureService default renderer wins the init race. Both compile fine. Caught only by looking.

FAILURE MODE 04

Environment trap.

Placeholder keys in appsettings.Development.json silently override your real keys. The error points at the registration system, not the file. Spell it out in the spec.

REASONABLE DOUBT · 20 / 24

Let's be honest about scope.

What this is not.

NOT 01

Not a replacement for GIS knowledge.

You still need to know what projection you're in, what a good legend looks like, and why your join is wrong.

NOT 02

Not a replacement for developers.

It's a force multiplier, not a substitute. Hard problems still need engineers. So does production.

NOT 03

Not magic for messy data.

If your bridge inventory is a swamp of nulls and inconsistent units, the AI won't fix it. Clean your data first.

NOT 04

Not finished. Not even close.

What works today is not what worked six months ago. Plan for the floor to keep rising.

REASONABLE DOUBT · 21 / 24

The last 10% is still 90%.

A 1985 observation that AI hasn't repealed.

> *"The first 90 percent of the code accounts for the first 90 percent of the development time. The remaining 10 percent of the code accounts for the other 90 percent of the development time."*
>
> — TOM CARGILL · BELL LABS · 1985

WHAT AI DOESN'T FIX

- **Edge cases** the demo never hit
- **Auth, deploy, data** plumbing
- **Performance** at real-world scale
- **Judgment** about what "done" means

THE BENEFICIARIES · 22 / 24

Where you sit in the room.

Who this changes things for.

IF YOU ARE AN

Analyst

You get the map you actually pictured.

More control over your web presence. Fewer rounds of "no, like this." A prototype before lunch instead of next sprint.

IF YOU ARE A

Developer

You stop writing the same boilerplate.

Spend your time on the hard parts — performance, security, the stuff that actually needs a brain. Let the translator do the wiring.

IF YOU DECIDE THINGS,

Decider

You can prototype before you commit.

Hours, not sprints. Ten variations of the same dashboard, side by side, before you pick one and write the spec for real.

CLOSING ARGUMENT · 23 / 24

If you only remember one thing:

The gap between

"I know what I want this map to do"

and

"I have a working app that does it"

just got dramatically smaller — regardless of where you sit on the technical spectrum.

VERDICT · CASE CLOSED

THE VERDICT

Thanks for sitting through it.

Come find us at the dymaptic booth. Bring your clues, and we'll help you strategize and plan.

GEOBLAZOR

Open source, MIT

github.com/dymaptic/GeoBlazor

CLAUDE CODE

Free tier available

claude.com/code

REACH THE SPEAKERS

Tim & Holly

dymaptic.com

BOOTH

ITAG 2026

The Meadows, Altoona

Let's Talk →

dymaptic.com