AUTOMOTIVE GRADE LINUX | VERY GOOD VENTURES

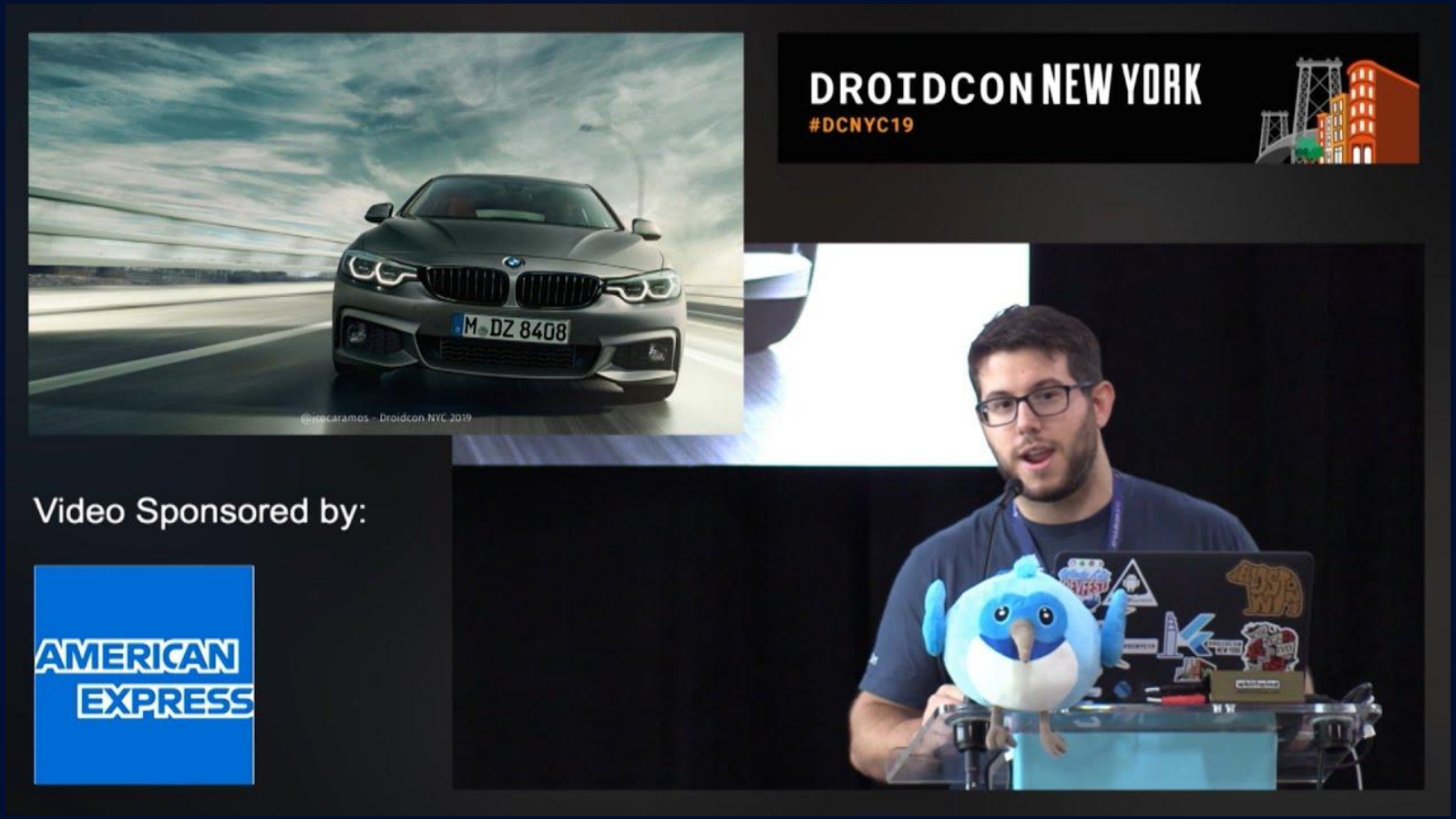
One codebase, every screen

Architecting multi-display Flutter experiences across automotive.

May 2026 - Jorge Coca, Head of Engineering

VERY GOOD VENTURES

I've been 10+ years obsessing with this ~~problem~~ opportunity



VERY GOOD VENTURES

VERY GOOD VENTURES

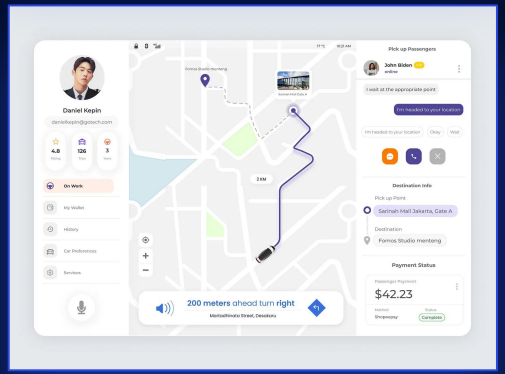
Why frontend is so fragmented in the vehicle industry?

VERY GOOD VENTURES

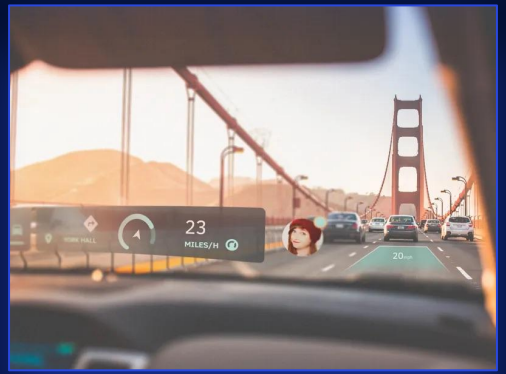
The car is already a multi screen device.

INSTRUMENT CLUSTER

IN-VEHICLE INFOTAINMENT

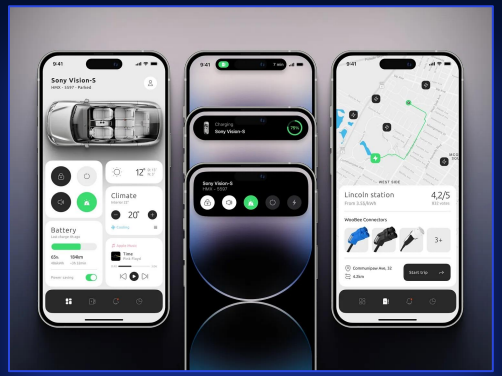


REAR DASHBOARD



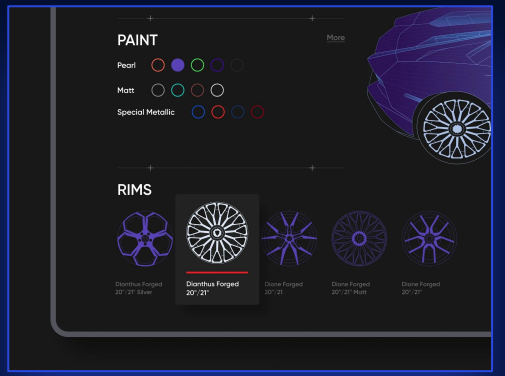
HEADS-UP DISPLAY

Outside the car, the industry is also multi screen



MOBILE APPS

EXPERIMENTS & SMART DEVICES



WEB CAR CONFIGURATOR

... and many more!

THE COST

The industry is paying for multi-screen multiple times

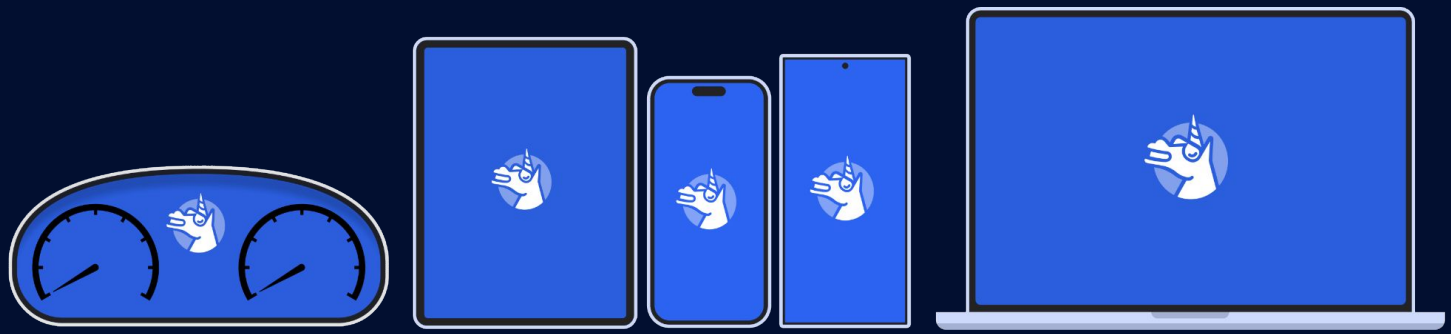
What the customer sees

- Cluster looks 2015. IVI looks 2025. Brand drift across surfaces.
- Companion app feels like a different product entirely.
- Dealer kiosk and service tablet share nothing with the vehicle.
- Innovation ships unevenly. One surface at a time.

What it costs the OEM

- 7+ surfaces. 7+ stacks. 7+ teams. 7+ release cycles.
- Same vehicle state model, reimplement 7 times.
- Same brand tokens, redrawn 7 times.
- Every change ships 7 times (or doesn't)

Flutter. This is the way!



VERY GOOD VENTURES

Flutter is an open source framework by Google for building beautiful, natively compiled, multi-platform apps from a single codebase.

Multiplatform

Only true multi-platform solution.

An efficient developer experience, scalable approach to testing and automation, and a platform portable codebase makes developers vastly more productive than developing with native tools.

Fast

Compiles natively to machine code, which enables fast app performance and high frame rates.

Unlike other frameworks that use a bridge and native components, Flutter's architecture guarantees the **highest performance possible** on every platform.

Productive

Development tooling is vast and extremely high quality, which improves velocity and efficiency.

Features like **Hot Reload**, VS Code extensions, and a robust, high quality community ecosystem enable Flutter developers to be extremely productive.

Scalable

Control over every pixel and modularity makes it easy to manage and scale.

Flutter gives developers complete control over the user experience, while also providing helpful tools to accelerate UI development. The ability to seamless build and maintain design systems across touchpoints accelerates velocity while maintaining quality.

How do we make it happen?

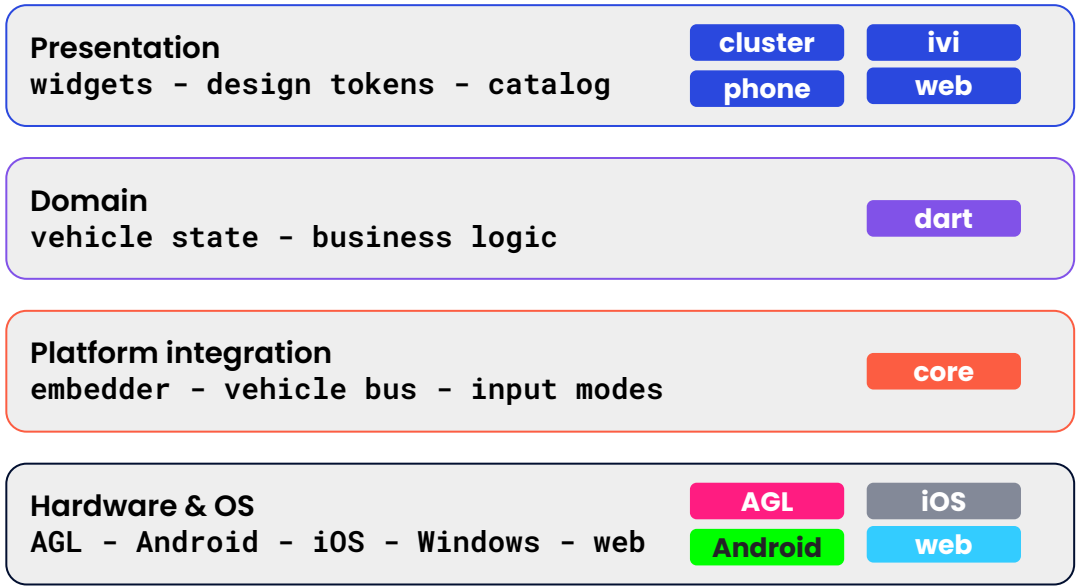
THE MENTAL MODEL

~~Write once, run everywhere.~~

Share the layers that should be shared. Isolate the layers that should not.

THE ARCHITECTURE MAP

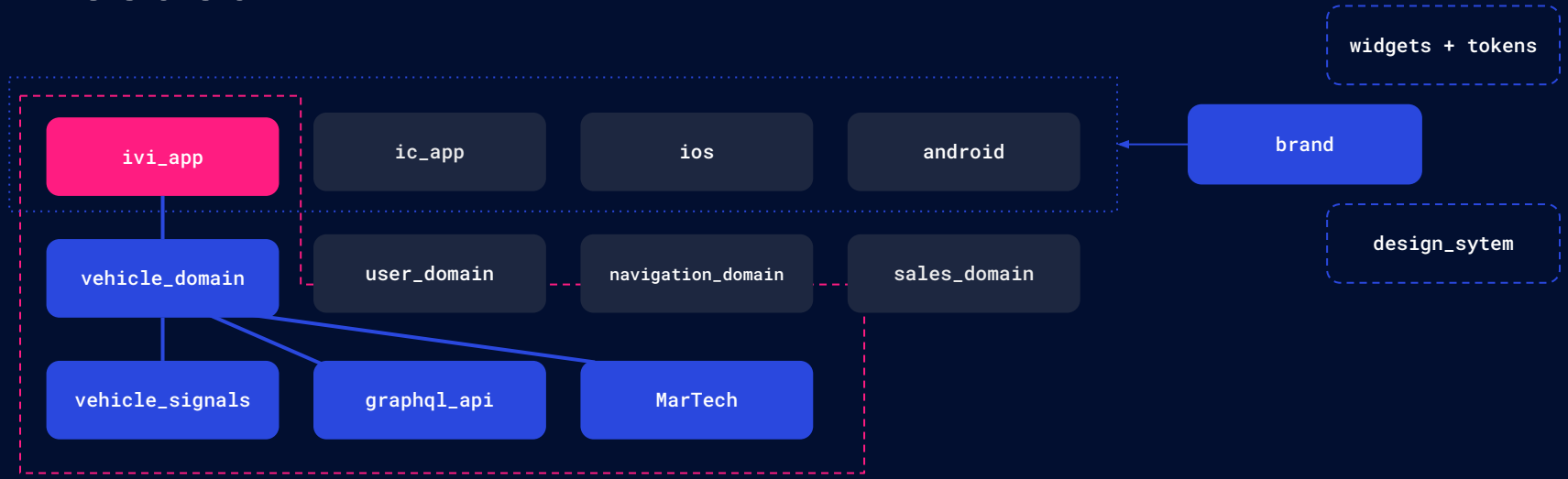
Four layers. Different sharing contracts.



Reading the stack

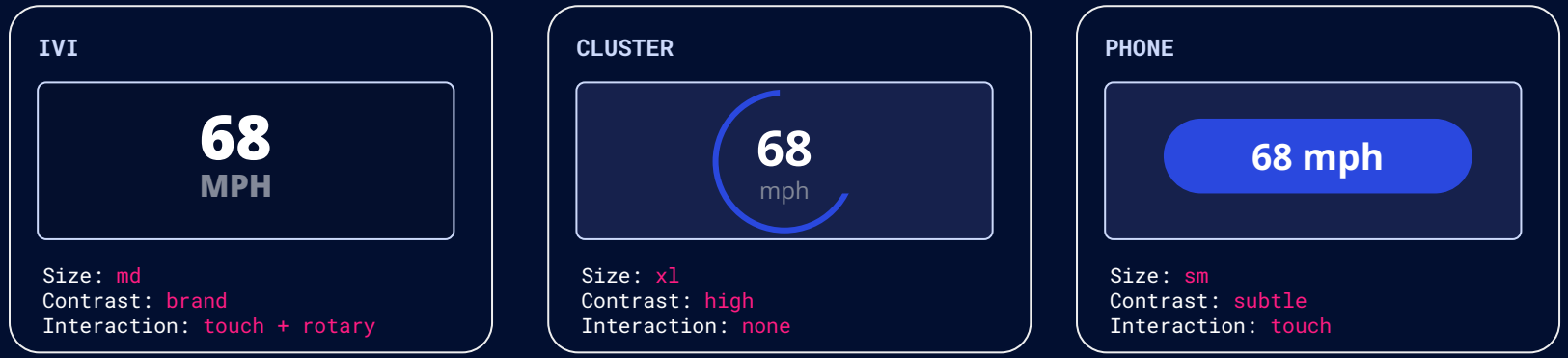
Presentation is shared at the widget/molecule and component level.
Platform mostly diverges, but can be shared (eg. authentication)
Hardware is whatever the surface runs on.

Multiple apps. Share code when, and where needed



For any platform, you compose your application as if it was a puzzle. You can either share and re-use models and modules, or create your own when needed

Same widget. Different tokens.



One widget. Three tokens. Three behaviors. Zero per-surface branches in the widget consumption itself. Brand sits on top as the last token later, and the system carries it.

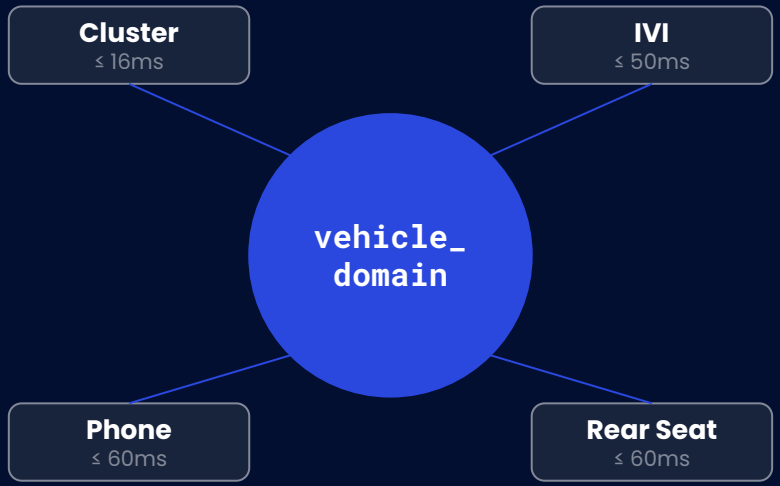
speedometer.dart

```dart
class Speedometer extends StatelessWidget {
  const Speedometer({
    required this.speed,
    required this.units,
    super.key,
  });

  final int speed;
  final String units;

  @override
  Widget build(BuildContext context) {
    switch (context.platorm) {
      case .iOS, .android:
        return _MobileSpeedometer(speed, units);
       case .agl:
        return _AglSpeedometer(speed, units);
       default:
        throw NotImplementedException();
    }
 }
}
```

One source of truth. No surface talks to another surface.



Surfaces subscribe to typed state streams

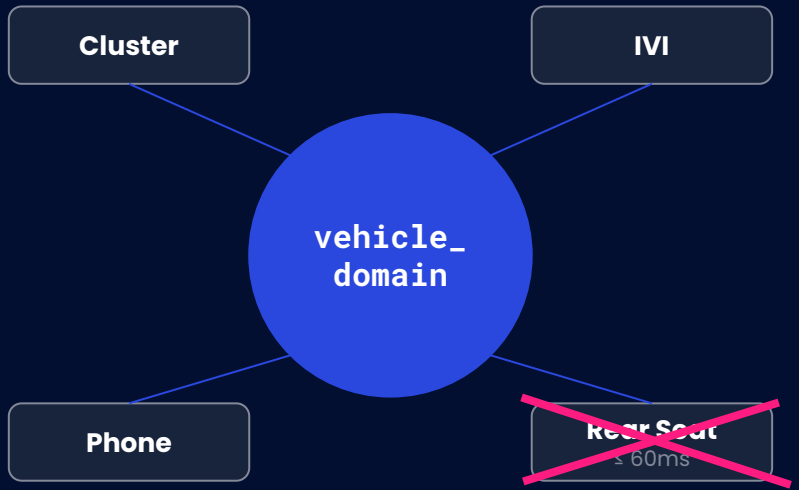
Each subscription can have its own latency budget

No cross-surface API. No direct calls.

Add a field to the model – every subscriber sees it on the next emission. No coordination required.

All written in Dart. Zero UI. Fast compilation. One source of truth that every surface reads from it.

Surfaces fail. The system keeps working.



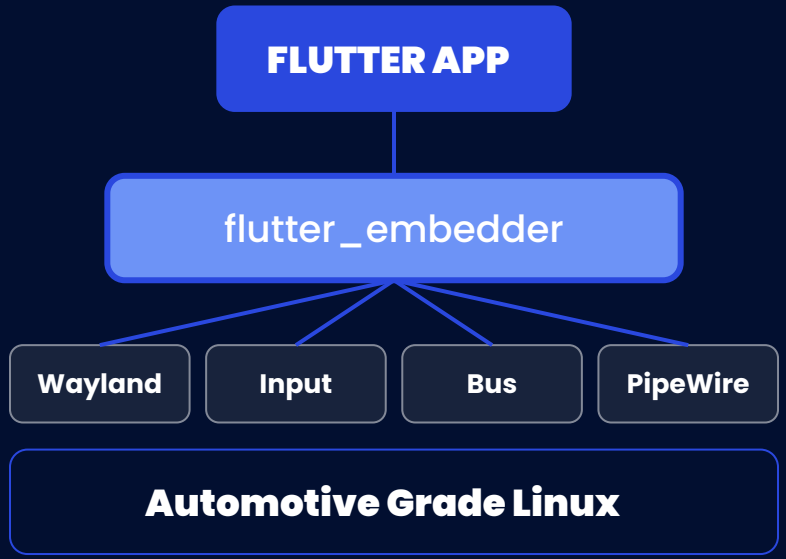
Cluster reboots. IVI keeps working.

Phone disconnects. Head unit unaffected.

Rear seat asleep. No errors propagate.

Surface comes back → resubscribes, catches up. No reconciliation protocol → there's nothing to reconcile.

Platform integration. Where Flutter meets AGL.



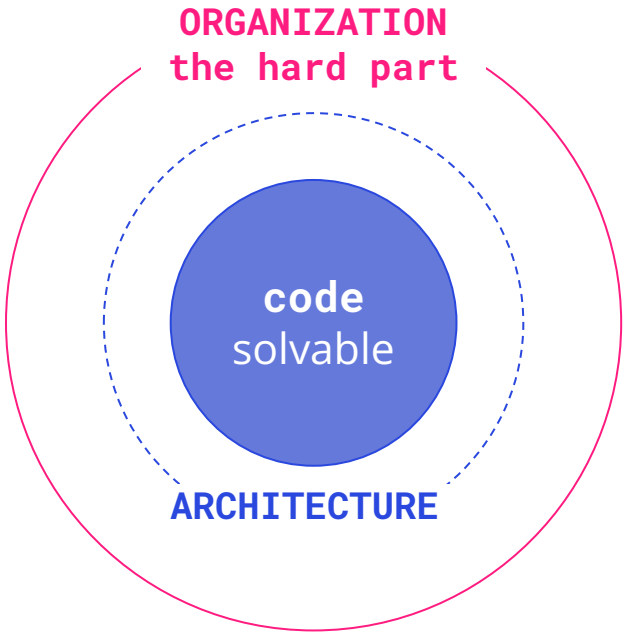
Inputs normalized. Touch, rotary, voice, hard keys arrive as a typed event stream. Widgets don't know which.

Vehicle signals abstracted. CAN, SOME-IP, VSS — all surfaced as typed Dart streams. Sim signals on the Pi today.

Contained. Differences live here so they can't leak upward.

Is it that easy? Where's the trick?

The hardest layer. Organizations do not have compilers.



Shared code requires shared planning. Product, design, engineering across surfaces.

The architecture exposes the org chart (Conway's Law)

> The technology works. The organization is the hard part. Naming this protects you from concluding that *Flutter didn't work* when really the org didn't.

AGL developer experience isn’t at mobile parity yet.

MOBILE FLUTTER TODAY

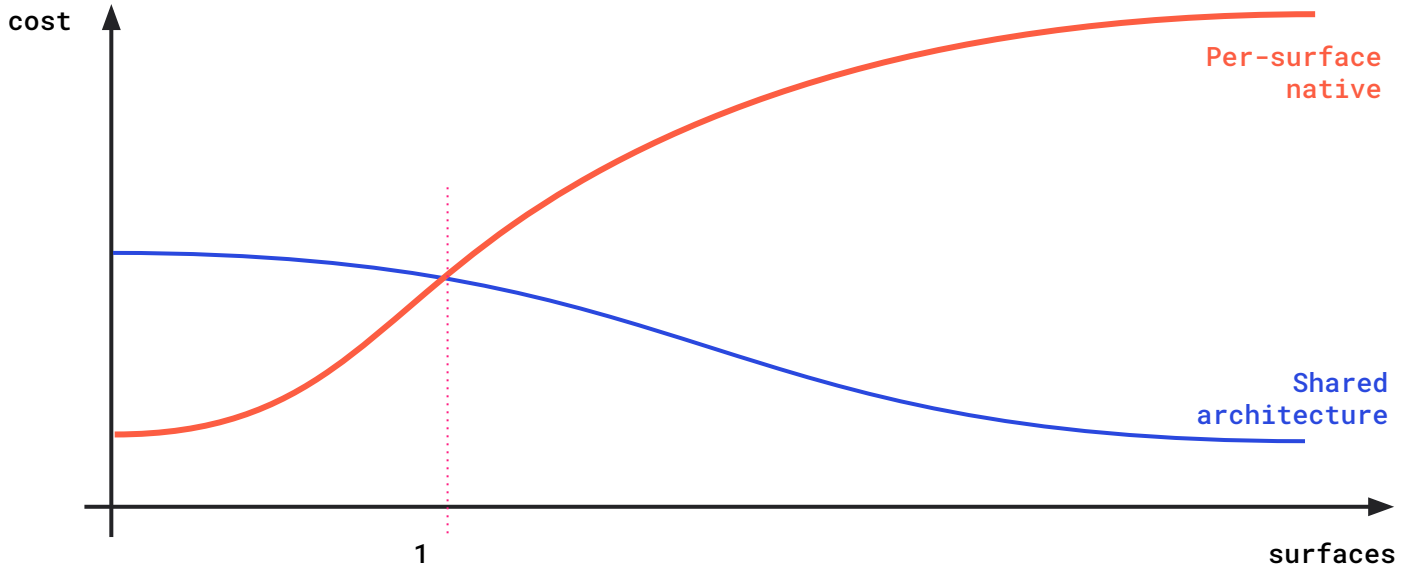
Hot reload in milliseconds
Instant deploy to devices
Mature embedder, large community
Worked examples for everything

FLUTTER ON AGL TODAY

Hot reload is not easy to replicate
Manual or scripted deploy to head unit
Embedder coverage is improving but uneven
Smaller community

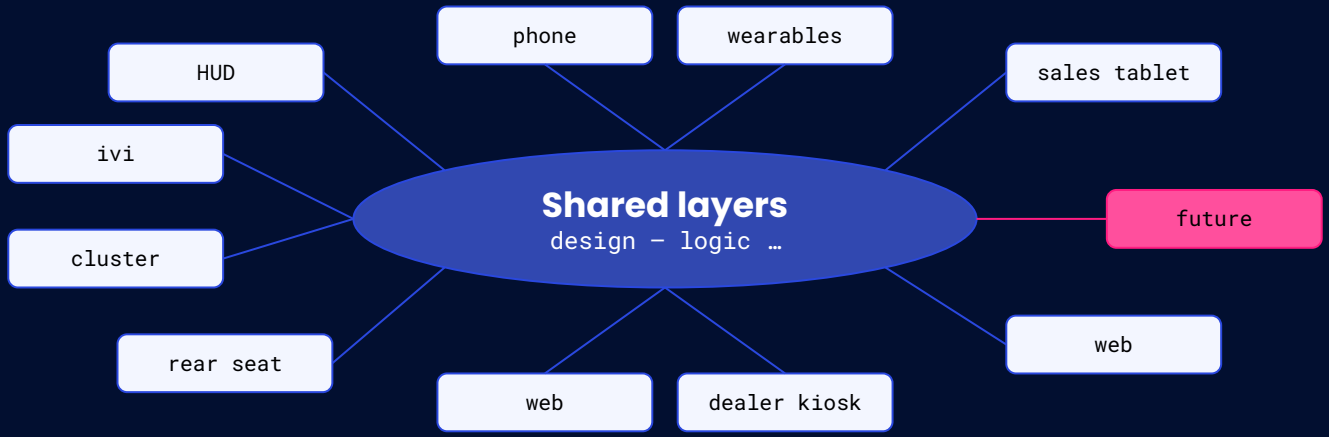
Eliminating friction out of the development process sparks creativity, standardizes the process, and invites contributors from all backgrounds to bring their ideas forward.

The investment is frontloaded.



With two or more, this is the cheaper option — and the more surfaces you add, the further it pulls ahead.

The same architecture is your entire digital stack.



Ship faster. Higher quality. Lower cost. Innovate more.

Ship faster

"One team". One design system. Shared roadmaps. New surfaces in sprints, not quarters or years.

Higher quality

Shared domain means surfaces can't drift. State is consistent by foundation.

Lower cost

High % of reusability. New surfaces are dramatically cheaper than the first one.

Innovate more

GenUI on day one, not quarters. Ready for the next interaction model on every surface at once.

These aren't promises. They're direct consequences of the architectural decisions covered in this talk.

How does it look in practice?

VERY GOOD VENTURES

How can you
best reach them?

Success! Flutter enables a single front-end technology across our digital stack

Multiplatform by design

Code is portable across operating systems. Siloed knowledge no longer needed.

Highest level of performance

60/120 fps at your fingertips, even on low-end HW. Native compilation. Small binaries.

Innovate. Experiment. Scale.

Significant reduction of time to market, even with the most original and imaginative use cases.

Smart ROI. Invest where your value is

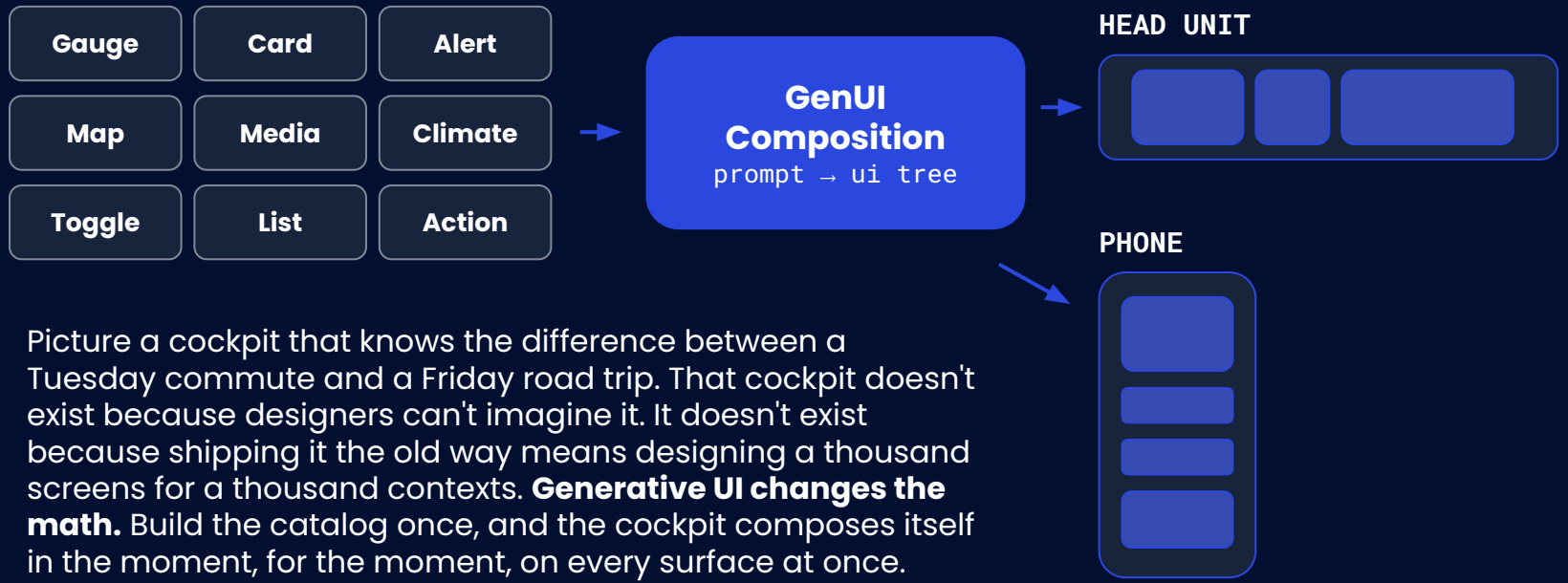
These aren't promises. They're direct consequences of the architectural decisions covered in this talk.

One more thing...

VERY GOOD VENTURES

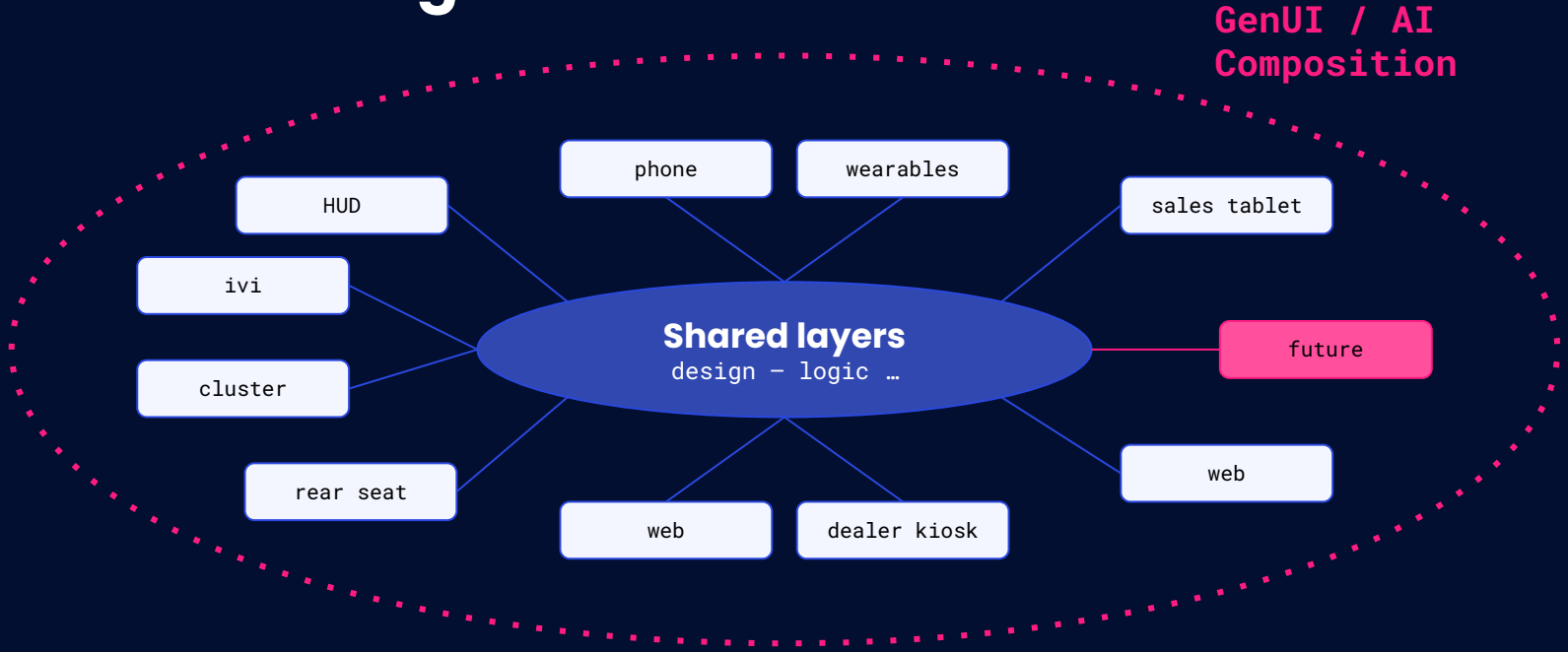
This would not be a tech talk if we don't talk about AI...

The widget catalog is also the GenUI catalog. AI draws the UI based on context and intent.



Picture a cockpit that knows the difference between a Tuesday commute and a Friday road trip. That cockpit doesn't exist because designers can't imagine it. It doesn't exist because shipping it the old way means designing a thousand screens for a thousand contexts. **Generative UI changes the math.** Build the catalog once, and the cockpit composes itself in the moment, for the moment, on every surface at once.

The same architecture is your entire INNOVATION digital stack.



WHY WE CAME TO TOKYO

AGL is the most serious open automotive platform in the industry.

Flutter is the most productive UI toolkit for multi-surface products.

The combination is underused, and we are here because we want to help change that.

We are investing in AGL.
We want to work with the people in this room who are trying to raise the UI/UX ceiling of their vehicles.

VERY GOOD VENTURES

Thank you

We’re excitect to connect with you.

hello@verygood.ventures
@VGVentures