



# THE FUTURE OF DATAFLEX

THE ROAD BACK TO  
EVERYWHERE

**LET'S RETRACE OUR STEPS OF THE  
ROAD WE HAVE BEEN ON...**

**... AND PUT THE PIECES TOGETHER  
THAT SHAPED DATAFLEX!**

# THE ROAD THE FIRST STEPS

## Data Access is born in 1976.

Before DataFlex was a product, Data Access was already solving business technology problems. That matters: the product's DNA starts with business software, not language theory.

### Origin story

Miami, business systems, and a company that learned by building practical solutions.

# A language shaped by business work.

DataFlex was formally presented at COMDEX in 1981 and developed in the early 1980s as a high-level 4GL for business applications.

D a t a F l e x  
Version 2.0  
Application Development Software System

Copyright (c) 1983, 1984 Data Access Corporation  
DATA ACCESS CORPORATION  
8525 SW 129 Terrace  
Miami, Florida 33156  
USA

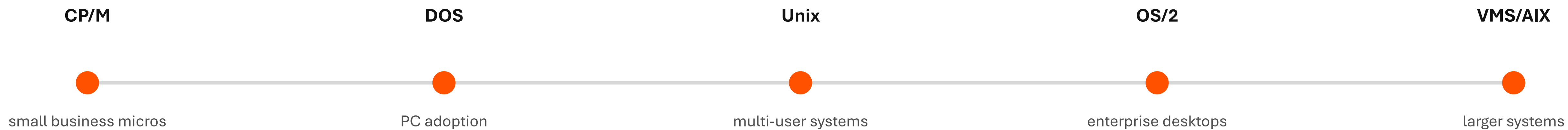
Manual Revision Date: 07/18/84

**THE ROAD**  
**THE FIRST STEPS – OUR FIRST PUZZLE**  
**PIECE**



# THE ROAD PORTABILITY

In the 1980s, DataFlex ran across the kind of machine diversity that business software had to survive: CP/M, MS-DOS, Unix, OS/2, VMS, AIX and others.



# ONE CODEBASE. MANY MACHINES.

That was a radical promise for business developers: write the application around the business, then take it where the customer needs it.

## THE ROAD

BY THE EARLY 90S, DATAFLEX WAS NOT AN EXPERIMENT.

Version 3.0 arrived in 1991 with a modernized interface, while DataFlex 2.3/3.x character-mode systems became known for reliability and long-lived business deployments.

**3.x**

character-mode longevity

**2.3**

older data compatibility

**THE ROAD  
OUR NEXT PIECE!**

## **Built to last**

Reliability across versions and systems



Compatibility that protects  
long-lived applications

**Longevity • Stability • Compatibility**  
Runs on many systems

# THE ROAD 3.X WAS A BIG ONE!

## Data Dictionaries

Validation, relationships, and security live in one consistent place.

## Data-aware UI

Controls understand the application's data model.

## Readable objects

UI-driven object notation is direct and approachable.

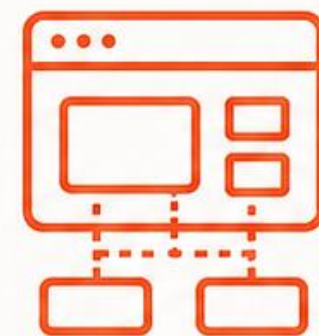
## One environment

Language, Studio, compiler, debugger, framework.

# THE ROAD 3.X WAS A BIG ONE!

## Objects built the UI

3.x introduced an object model ahead of its time



Nest objects, set properties,  
and the interface takes shape

Object model • UI by composition •  
DataDictionaries • UI bound to data

# THE ROAD JUNCTION DATAFLEX AND CUSTOMERS CHOSE WINDOWS!

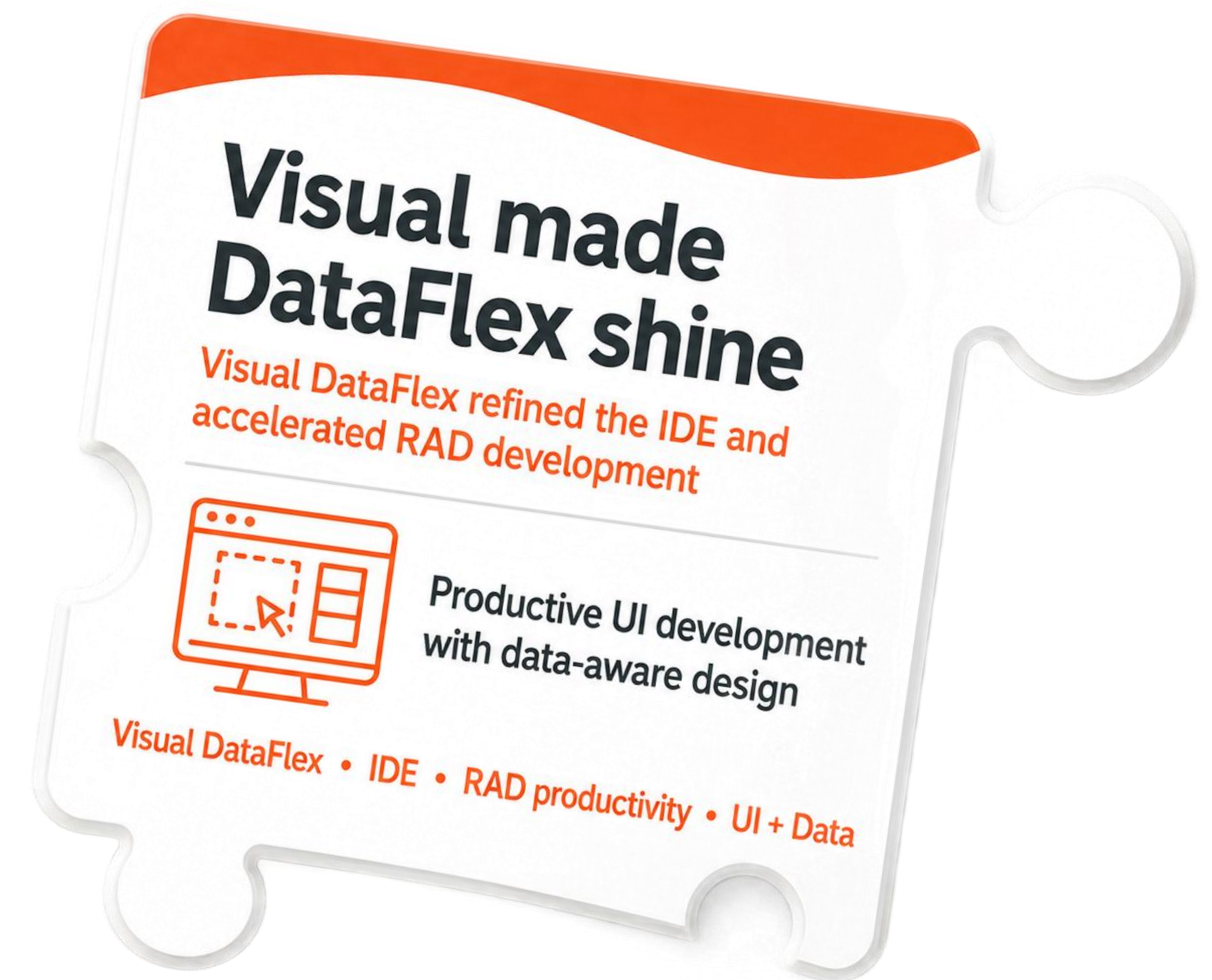
## DataFlex needed focus!

Markets moved to Windows as not only businesses did but also consumers with the spread of the internet and a more user-friendly PC!



# THE ROAD THE WINDOWS ERA GAVE DATAFLEX A FACE

From version 4 in 1996 through version 17.1, the product was branded Visual DataFlex: a GUI development language and IDE for Windows business applications.



# THE ROAD USERS MOVED FROM WINDOWS TO BROWSERS

The next shift was not just a new UI. It was a new deployment model: central servers, remote access, browsers, mobile devices. The World Wide Web grew up!



# THE ROAD

## 2012: A BRIDGE TO THE WEB

Visual DataFlex 2012/17.1 introduced a Web Framework for transitioning Windows business applications to the web, allowing developers to express applications in traditional Visual DataFlex code instead of spreading logic across ASP, JavaScript, HTML, and CSS.

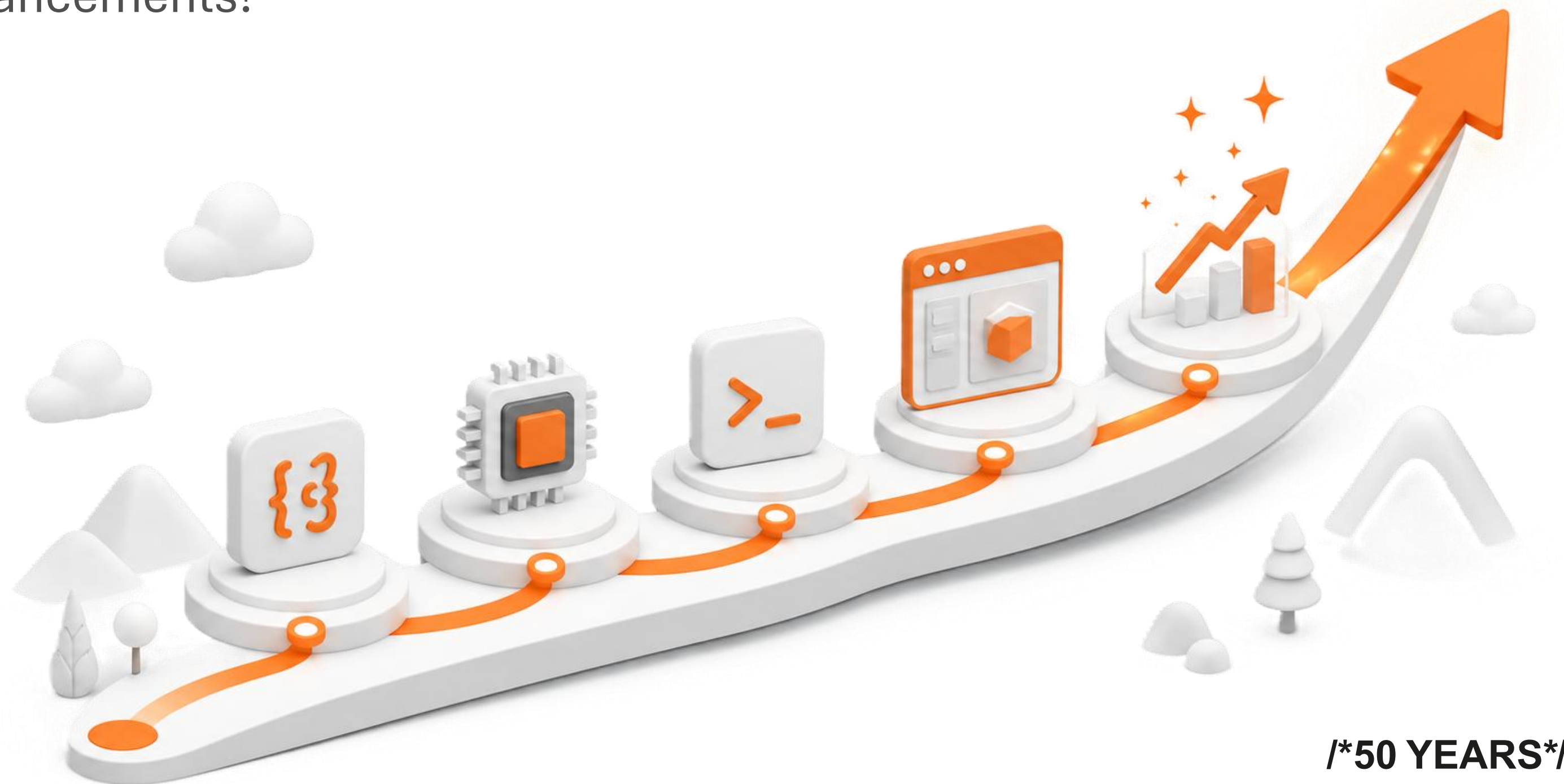
It came at a cost of rewriting the user interface, but our DNA remained the same which allowed you to do what you do best!

### **A familiar path**

DataFlex code, Web Objects,  
Studio preview, Focus on  
productivity!

# THE ROAD 2019-2026

DataFlex continued that trend by moving developers and their applications to the next big things and satisfy the developers needs to stay competitive with 20.0's Unicode & 64-bit, 23.0's SQLExecutor & FlexTron, and 24.0- & 25.0's productivity enhancements!



# THE ROAD THE FINAL PART OF THE PUZZLE FOR NOW

## Developers moved forward too

DataFlex kept bringing developers  
to the next platforms



From one generation of platforms to the next

Windows • Web • FlexTron • What comes next

**THAT IS EXACTLY WHAT IS  
HAPPENING AGAIN!**

**BECAUSE AROUND THE SAME TIME...**

**ANOTHER JUNCTION APPEARED!**

# ANOTHER JUNCTION NEXT GENERATION OF APPS

Now the client is powerful. Offline is expected. The cloud is normal. As such we are at another crossroads! DataFlex is not as portable anymore as it used to be!

**The problem is not that DataFlex can't move forward. It is just that we followed the road into a building with too many Windows in it!**



# ANOTHER JUNCTION NEXT GENERATION OF APPS

Today, business apps are judged against applications that run anywhere: that have local experiences, instant UI, sync later, API-first integration, elastic hosting, containers, and cross-platform operations.



## OFFLINE → SYNC → VERIFIED DATA

- The repairman story is simple: store changes temporarily on the device, then sync to the server when the connection returns. This is the customer-facing proof point.



**WE ARE ALREADY ABLE TO DO THIS  
MOSTLY USING WEB TECHNOLOGY!**

# THE ROAD SERVER-HEAVY BY DESIGN.

In the current model, most interactions ask the server: what should happen now? This is reliable, but it makes offline and highly responsive local interaction hard.



click / save / verify / store

# THE ROAD TO TAKE THE MARKET EXPECTS A FAT CLIENT.

Interactive logic can run locally. Shared data and authoritative rules remain server-side where trust and synchronization matter.



**AN APP THAT RUNS ON ANY DEVICE,  
WITH A LOCAL EXPERIENCE! AND  
THAT IS WHAT YOUR CUSTOMERS  
EXPECT!**

**THE ROAD TO TAKE  
PREFERABLY ON ONE OF THESE!**



# DECISION THE TECHNOLOGY STACK

At the last conference(s) we introduced to you the new technology stack. A way to make this all possible with one simple principle in mind, but in other words keeping the pieces of our DNA in a changing world!

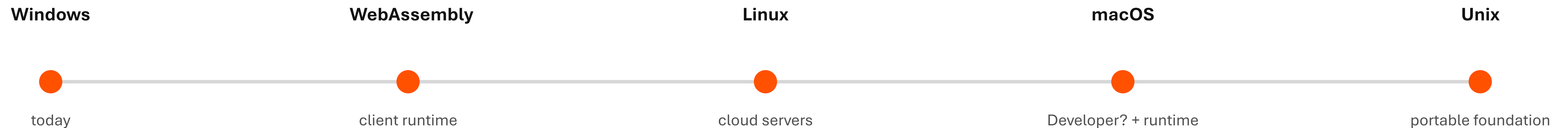
## Principle

Rebuild what limits the future.  
Preserve what defines DataFlex.

# RUNTIME OR TO SOME VIRTUAL MACHINE

## A PORTABLE CORE

The old path optimized for Windows. The new path starts with portability as a core requirement: browser via WebAssembly, server on multiple platforms, and a more platform-agnostic API surface.



# TECH STACK MEANS: DATAFLEX REBUILT FOR THE NEXT DEPLOYMENT ERA.

Not a new language. Not a cosmetic release. A new foundation that lets the ecosystem target browser, server, cloud, containers and operating systems again.

# WE PROTECT WHAT WE STAND FOR AND WHY YOU CHOSE US!

## Developers moved forward too

DataFlex kept bringing developers  
to the next platforms



From one generation of platforms to the next

Windows • Web • FlexTron • What comes next

## Visual made DataFlex shine

Visual DataFlex refined the IDE and  
accelerated RAD development



Productive UI development  
with data-aware design

Visual DataFlex • IDE • RAD productivity • UI + Data

## Not just code

A language for business processes

 Close to the database

Rules • Screens • Reports • Productivity

## Built to last

Reliability across versions and systems

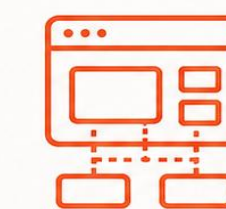


Compatibility that protects  
long-lived applications

Longevity • Stability • Compatibility  
Runs on many systems

## Objects built the UI

3.x introduced an object model ahead of its time



Nest objects, set properties,  
and the interface takes shape

Object model • UI by composition •  
DataDictionaries • UI bound to data

# THIS IS TECH STACK'S FIRST STATUS UPDATE!

# SCOPE

## IT TOUCHES EVERY PART OF THE ECOSYSTEM

### Runtime

Built from scratch

### Compiler

Updated and extended

### Expression Compiler

New

### Portable APIs

Reworked

### Application model

New

### Debugger

Native + Chromium

### Interop

Modernized

### Studio

Deep integration

### Packages

Ecosystem shift

# STATUS

## SCOPE OF THE PROJECT

- 700,000 LOC changed since Croatia in Tech Stack alone!
- Studio, Modules, ...
- Not including mainstream development (24.0-26.0)

**THANKS TO ALL THE HARD WORK  
FROM THE ENTIRE TEAM WE GAVE  
OUR FIRST TECH STACK TRAINING!**

# WEBASSEMBLY

## THE BROWSER BECOMES A RUNTIME TARGET

WebAssembly is an instruction format supported by modern browsers. It makes it possible to compile a runtime to a compact, high-performance browser-executable format — which is why it matters for DataFlex on-device execution.

### Why it matters

DataFlex code can run in the local browser instead of round-tripping every interaction.

## WEBASSEMBLY ALLOWS MOBILE AS WELL

- During the training we built a fully capable mobile app for Android (and iOS in the future)
- By bundling the runtime, DataFlex application, and Web UI in an App bundle
- Which allows for the very same functionality; offline and a performant user experience without latency
- We will be providing a bundling service so that you don't need a Mac to create App builds; while you are fully able to do it yourself as well!
- Some worry about performance; but there is no latency (except on live data), precompiled WebAssembly of the runtime gets even further optimized by the browser and transpiled to native instructions

# DATAFLEX CODE, RUNNING ON ANY DEVICE USING WEB TECHNOLOGY!

That is the key mental model for Phase 1. The browser is not just a renderer; it becomes an execution environment for DataFlex logic

**SO, LET'S TAKE A LOOK!**

# NEW WEBAPP MODEL

## PROVE THE PORTABLE FOUNDATION FIRST

### WebAssembly runtime

DataFlex logic runs in the browser.

### Native Windows runtime

Server-side and terminal foundation.

### Chromium debugging

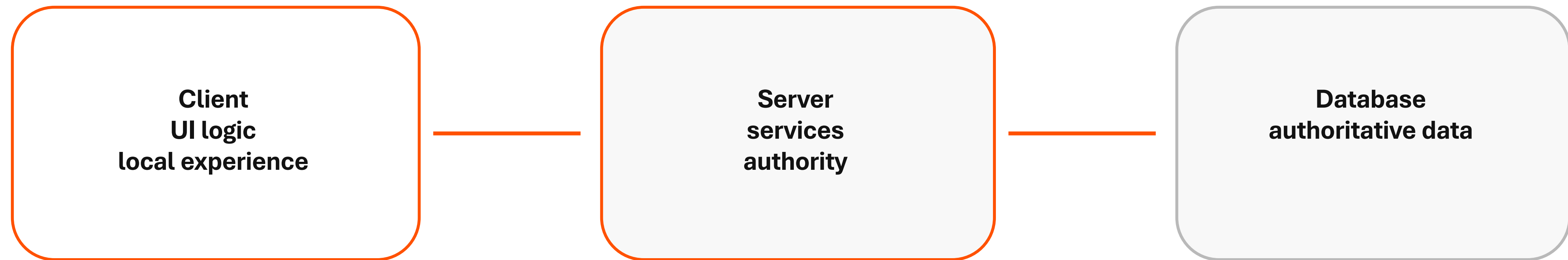
Debug the new browser runtime.

### Portable data foundation

SQLite-backed sample scenarios.

# NEW WEBAPP MODEL

## SPLIT THE WORK WHERE IT BELONGS



Local where responsiveness matters. Central where authority, persistence, integration and shared data matter.

**THERE IS ONE COMPONENT AT THE  
HEART OF ALL OF THIS...**

**THE STEPPINGSTONE OF THE  
PACKAGE MANAGER GAVE US AN  
ENTIRELY NEW BUILD SYSTEM!**

# THE DYNAMIC BUILD SYSTEM IS THE NEW CORE THAT CONNECTS IT ALL!

- df-cli that sits at the core of the studio
- In the new tech-stack release even more fundamental
- As it handles different toolchains
- Makes the entire ecosystem more dynamic and modular switching out components when needed
- Handling different:
  - Runtimes
  - Compilers
  - Debuggers



# THE DYNAMIC BUILD SYSTEM IS THE NEW CORE THAT CONNECTS IT ALL!

- It's how we handle the new WebAssembly-based platform
- Next to the existing 32-bit and 64-bit runtime, compiler, and debugger
- Both DataFlex Windows and the New DataFlex
- How we'll support DataFlex Windows in the long run as well!
- But this makes more sense with a demo!



# SHORT DEMO

**THIS IS ALL AMAZING STUFF...**

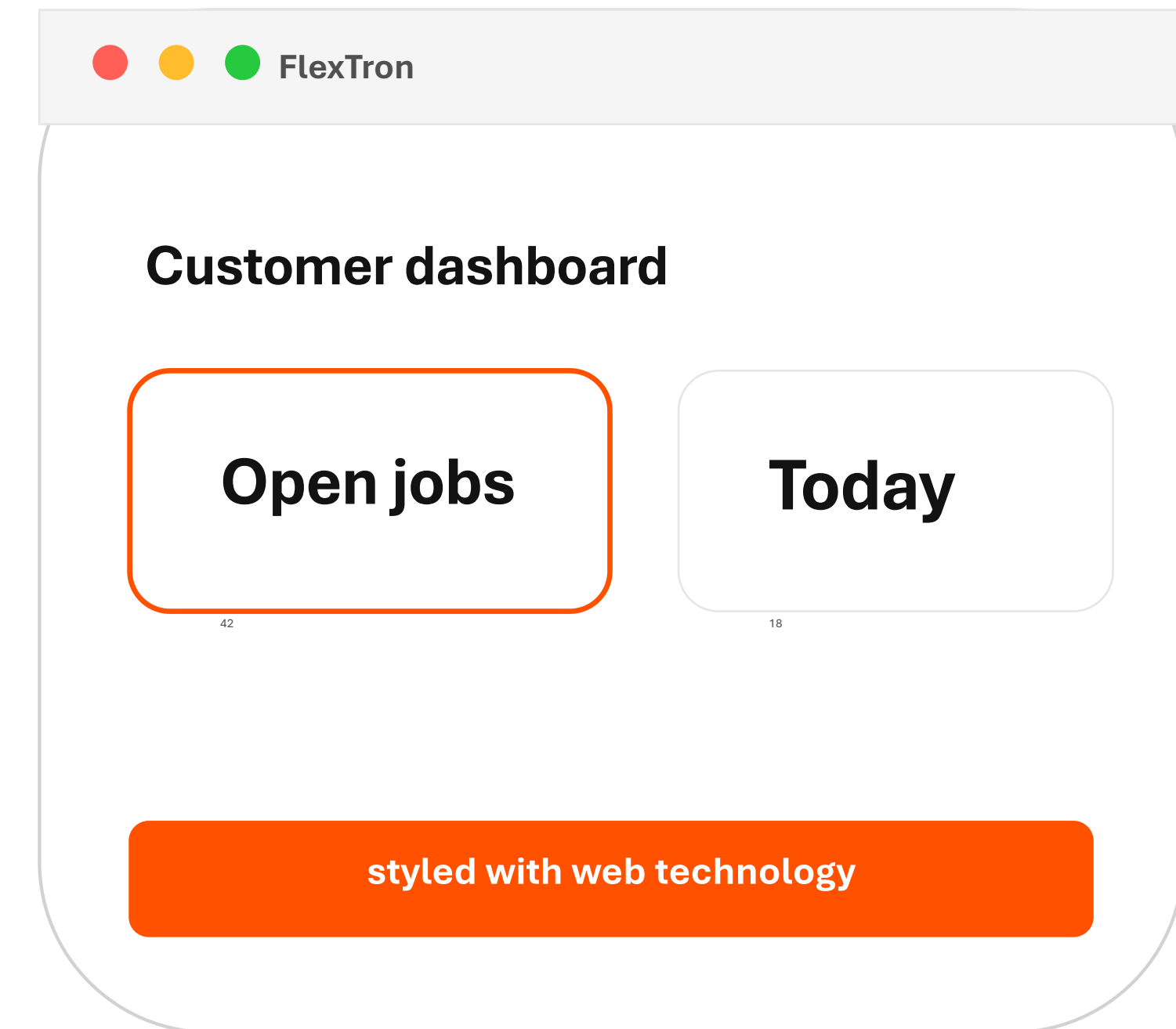
**BUT THERE SEEMS TO BE A  
MISCONCEPTION...**

**THAT IS THAT DATAFLEX IS NOT  
FOCUSING ON, NOR SUPPORTING  
DESKTOP APPS ANYMORE....**

# DESKTOP APPS ARE NOT DISAPPEARING THEY ARE CHANGING SHAPE

## Desktop apps are not disappearing. They are changing shape.

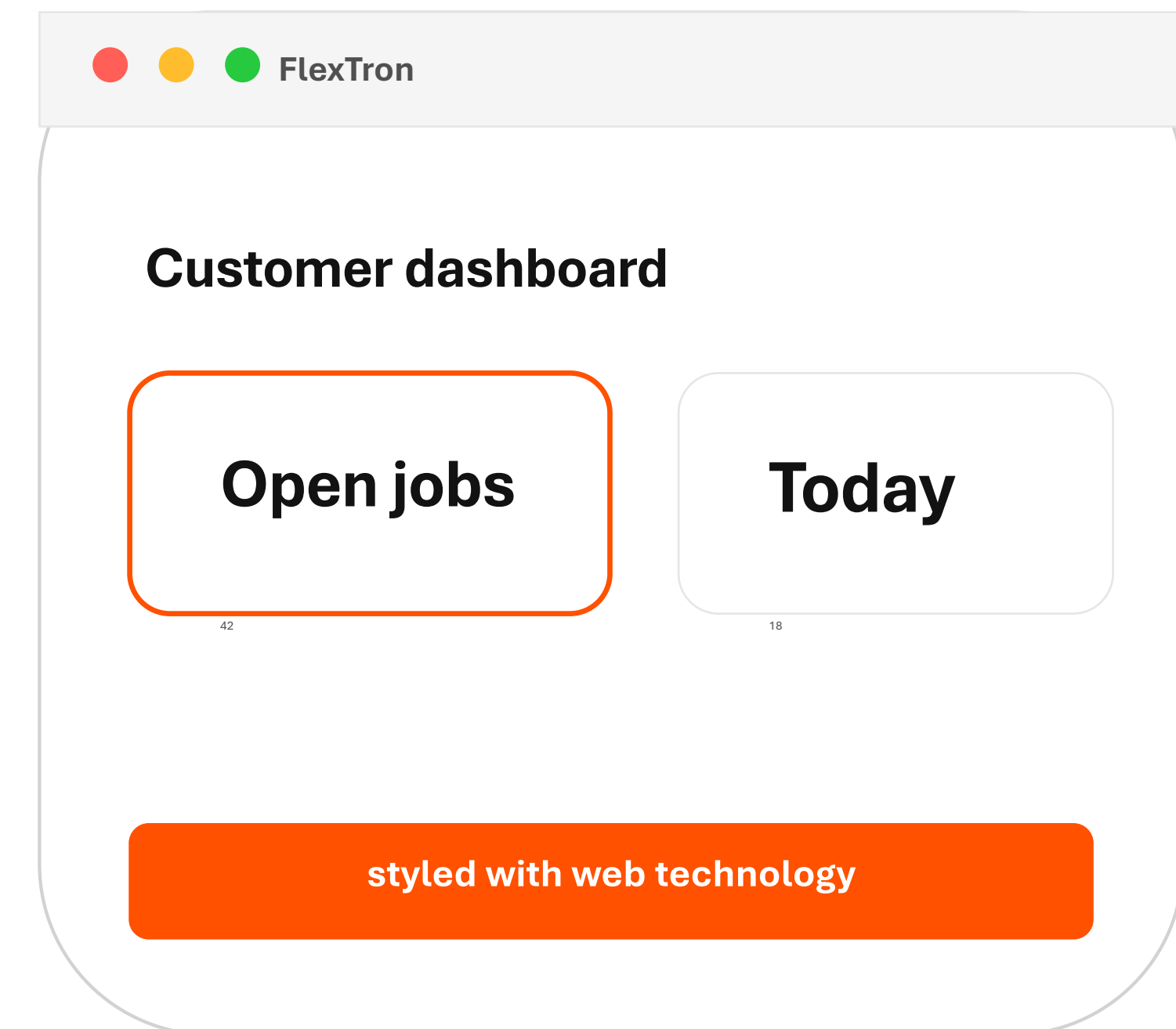
Tech Stack does not mean “everything must run as a web application” It also keeps the desktop-application story alive through WebView/FlexTron/Electron-style patterns.



# FLEXTRON

## WEB TECHNOLOGY INSIDE A CLIENT APPLICATION.

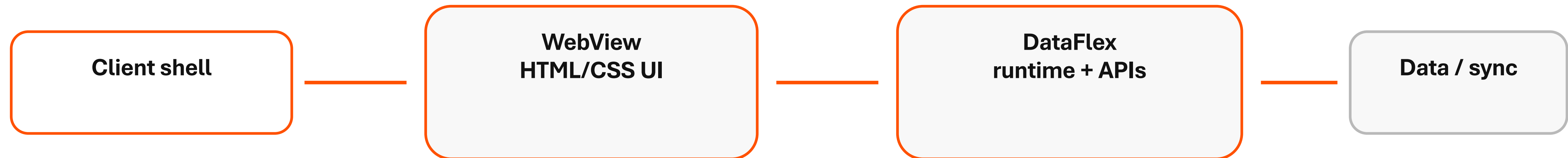
FlexTron allows the usage of DataFlex web controls within Windows desktop applications. It is a way to create nicer-looking UIs using web styling options, as a migration path from Windows to web, and as an alternative to ActiveX controls.



# FLEXTRON MENTAL MODEL

## THINK ELECTRON-STYLE, BUT DATAFLEX-FLUENT.

The valuable idea is not the brand name Electron. It is the pattern: a native client shell, a web-rendered UI, CSS-level styling freedom, and application logic that can stay close to the DataFlex world.



# WHY IT MATTERS IT PROTECTS THE DESKTOP STORY

## Modern UI

Use web styling to make applications feel current.

## No browser-tab requirement

A deployed client application can still be the right product shape.

## Migration bridge

Teams can introduce web views and controls before committing to a full web rewrite.

## Technology continuity

DataFlex developers keep building business applications with familiar concepts.

**TODAY WE ARE DEMONSTRATING  
THAT FACT!**

# PORTABILITY PROVEN

- By starting today on your Flextron journey
  - Better looking apps
  - Style it how you want potentially with a bit of help from a lovely chatbot for those that like small forms 😊!
  - Gives porting opportunities to Web Apps, Mobile, and in the future other Desktops!
  - Guarantee the continuity of your application as the world moves on!



# ROADMAP DIRECTION

## MAKE DATAFLEX CLOUD-NATIVE ENOUGH TO CHOOSE YOUR PLATFORM

Phase 2' goal is a cross-platform WebApp server model with deployment patterns closer to the rest of the cloud world: Linux, containers, load balancing, Kubernetes/Docker, pre-built images, and lower-friction hosting choices.

### **Outcome**

Scale more easily. Host more flexibly. Reduce Windows/IIS lock-in.

# ROADMAP DIRECTION

## DATAFLEX ON ALL PLATFORMS AND DEVICES!



# ADOPTION DIRECTION

## A TRANSITION, NOT A CLIFF

Over time, the ecosystem can support several deployment styles while teams adopt Tech Stack at their own pace.

### **TS server + TS client**

Best full-stack integration.

### **TS client + REST**

DataFlex UI with any backend.

### **TS server only**

Modern runtime, familiar paradigm.

### **TS Cross Platform Desktop Apps using Flextron**

For those that want a Desktop experience

### **Current server + TS client**

REST-based bridge mode.

### **Current DataFlex Windows WebApp**

If it works, keep it supported.

# ROADMAP

## THOUGH NOT LOCKED IN PLACE WITH NEW VELOCITY

| Technology                 | WebAssembly | Windows | Linux | MacOS |
|----------------------------|-------------|---------|-------|-------|
| Runtime                    | 1           | 1       | 2     | 2     |
| Studio Debugger            | 1           | 1       |       |       |
| Web UI                     | 1           |         |       |       |
| SQLite                     | 1           | 1       | 2     | 2     |
| SQL Driver                 |             | 1       | 2     | 2     |
| FlexTron                   |             | 1       | 2?    | 2     |
| WebApp Server / IIS        |             | 1       |       |       |
| New Web Server             |             | 2       | 2     | 2     |
| Cross Platform Development | 3           | 3       | 3     | 3     |

```
IF (TRAINING_IS_SUCCESS());  
    FUNCTION_RETURN "PREVIEW SOON"
```

**WE INTEND TO RELEASE A TECHNICAL  
PREVIEW OF 27.0 AFTER FIXING SOME  
POST TRAINING FEEDBACK**

**WHICH HAS THE FOCUS  
WEBAPPCLIENTS WITH OTHER  
PLATFORMS COMING SOON AFTER**

# TECHNICAL PREVIEW

## FLEXIBLE ECOSYSTEM

- 27.0 is a small layer on top of 26.0 modularizing components and furthering toolchain switching
- We intend to take full advantage of the 26.0 package manager and do monthly updates
- Toolchains will in the future also be installed through the package manager but for now have separate installers every month
- We'll keep Studio updates to a minimum while we don't have a nice update mechanism yet

# TECHNICAL PREVIEW FLEXIBLE ECOSYSTEM

- During each month you can give your feedback on new features, bugs, features you are missing from DataFlex Windows
- We will open a forum for Tech Stack
- We might continue the technical preview even after a full release

# TO CONCLUDE WHAT CUSTOMERS SHOULD HEAR?

## Continuity

Your DataFlex investment has a path forward.

## Capability direction

Offline, browser-local logic, cloud deployment and cross-platform reach.

## Choice

Adopt gradually. Mix current and Tech Stack models.

## Confidence

Current runtime support continues for a considerable time.

# SO MUCH MORE TO BE SHARED

## OTHER PRESENTATIONS

- Today
  - **Fresh Eyes, real Changes: Evolving the Tech Stack** by Jip
- Tomorrow
  - **The New DataFlex Build System** by Bram
  - **No more global file buffers!** by Harm
- Next to all brilliant things presented on 26.0 the foundation for all of you'll be seeing
- Not to forget our guest presenters
- All part of the puzzle that is DataFlex!

# THE ROAD DOES NOT END AT WINDOWS. IT RETURNS TO EVERYWHERE.

DataFlex started with portability, survived by focusing, evolved through the web,  
and now rebuilds its foundation for the cloud and device era.