

DataFlex

TECH STACK
MEANS
BUSINESS

BRAM NIJENKAMP

0

BOB

CONTEXT
BOB



BOB

CONTEXT
BOB (THE REPAIRMAN)



BOB (THE REPAIRMAN)

CONTEXT
BOB (THE REPAIRMAN)



CONTEXT
BOB (THE REPAIRMAN)



CONTEXT

BOB NEEDS AN APP TO DO HIS WORK

An App that stores the changes on his phone temporarily

But then syncs his changes to the server once he has an internet connection

WE HAVE HEARD THIS STORY MORE
THAN ONCE....

HOW DO WE BUILD AN APP IN
DATAFLEX THAT WORKS OFFLINE...

WE CAN'T DO THAT YET?!

1

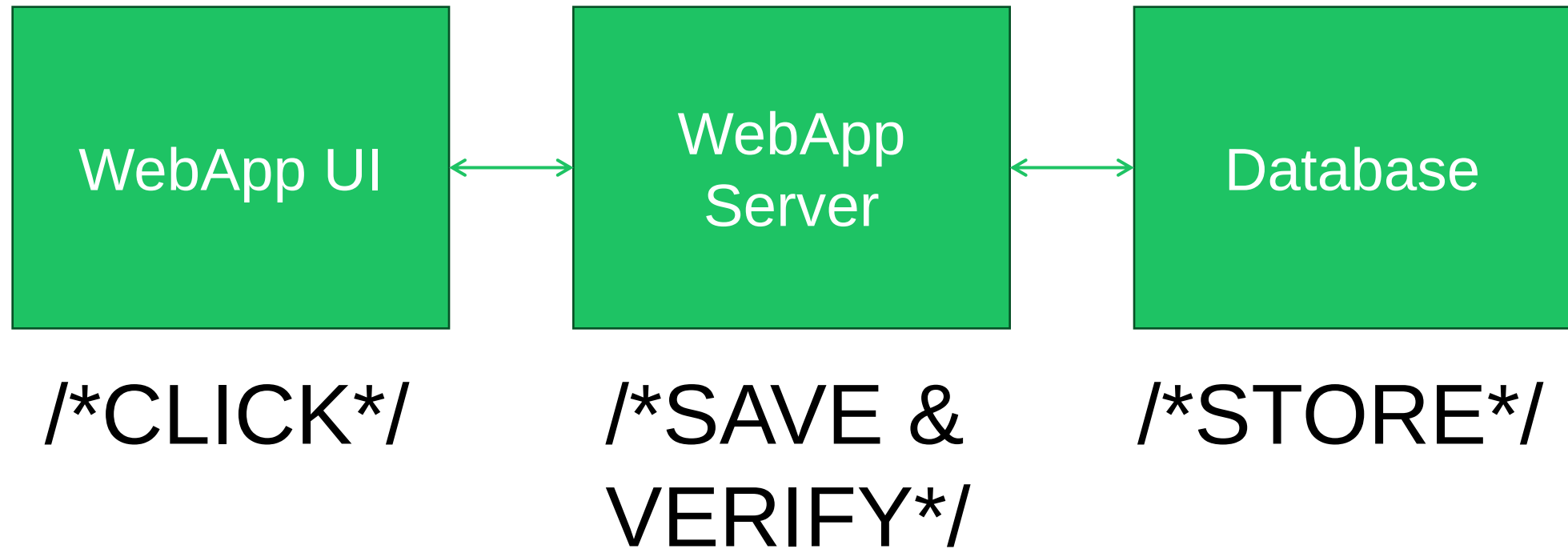
WHAT ARE DATAFLEX WEBAPPS?

CONTEXT

EXISTING WEBAPPS

Current DataFlex WebApps are server heavy

Meaning that every interaction goes to the server and asks: “What do I need to do with this?”



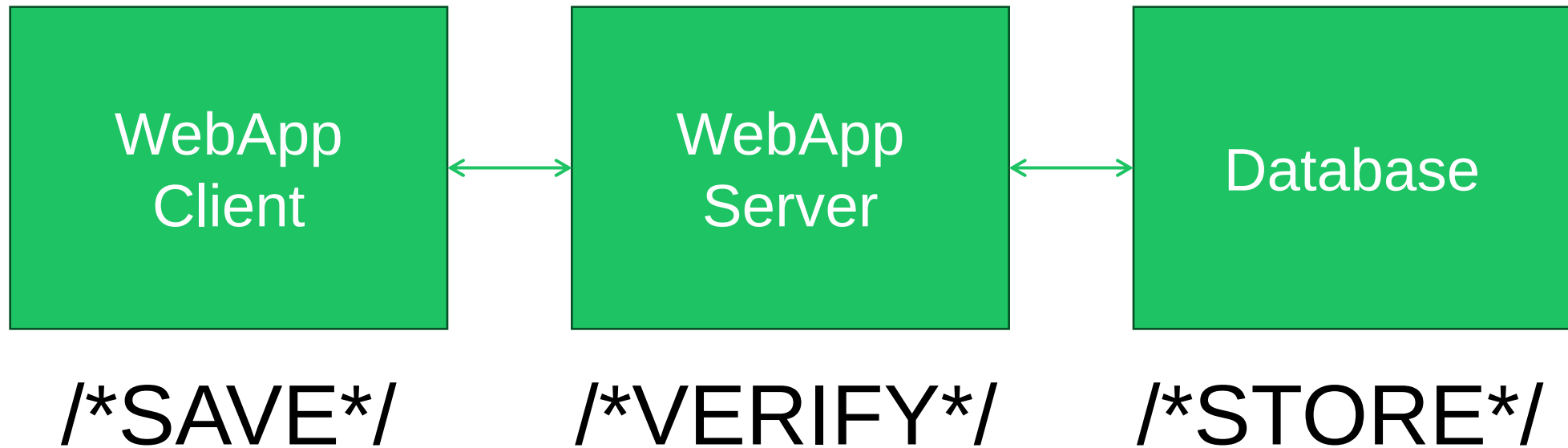
CONTEXT

PRESUMED WEBAPPS

Has a fat client

With all business logic in the browser or on device

Communicated only with the server when needed



THIS SEEMS COMPARABLE TO
OTHER LANGUAGES...

2

TECHNOLOGY STACKS

TECHNOLOGY STACKS

THE CHOICE IS YOURS!



Front-End
(Client)



Back-End
(Server)



Database



Apps



HTML



CSS



React



Angular



VueJS



Node



Python



Java



C#



Oracle



PostgreSQL



MySQL



SQL Server



IBM DB2



Android Studio



Swift



Xamarin

TECHNOLOGY STACKS

THE CHOICE IS YOURS!



Front-End
(Client)



Back-End
(Server)



Database



Apps



HTML



CSS



React



Angular



VueJS



Node



Python



Java



C#



Oracle



PostgreSQL



MySQL



SQL Server



IBM DB2



Android Studio



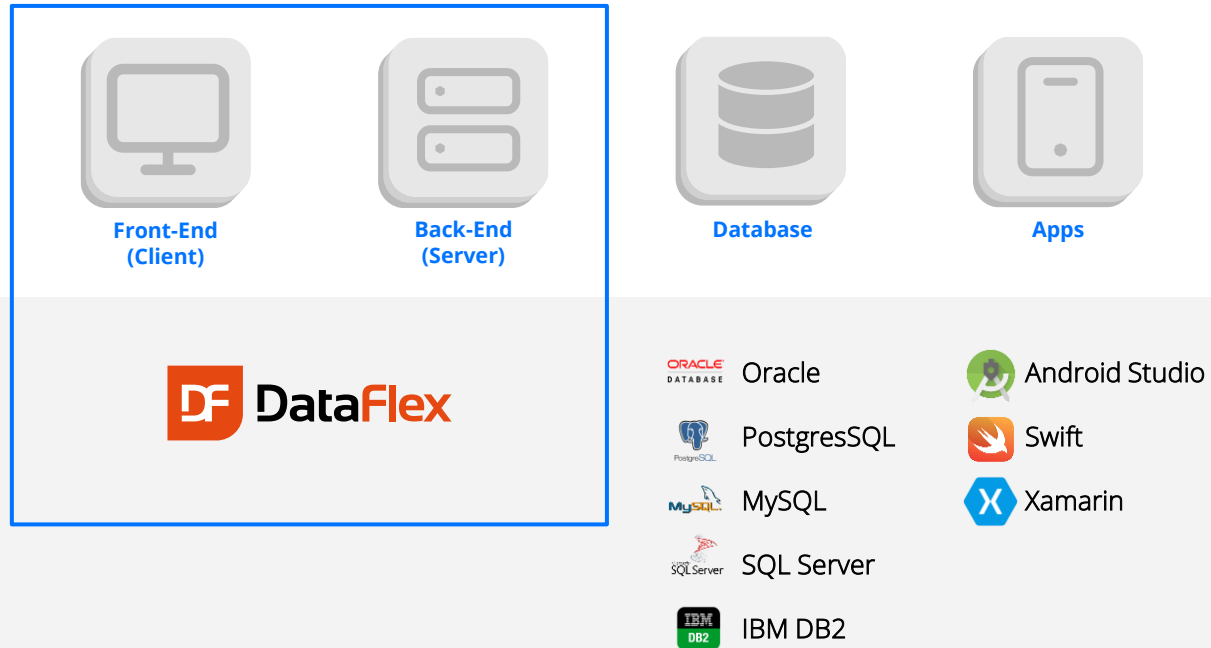
Swift



Xamarin

TECHNOLOGY STACKS

WHERE DOES DATAFLEX FIT IN?



TECHNOLOGY STACKS

WHERE DOES DATAFLEX FIT IN?



Front-End
(Client)



Back-End
(Server)



Database



Apps



Oracle



PostgreSQL



MySQL



SQL Server



IBM DB2



Android Studio



Swift



Xamarin



DataFlex

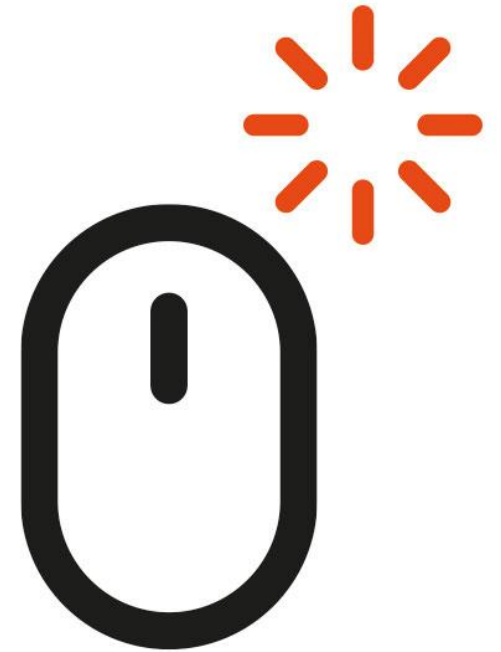
3

HOW ARE WE GOING TO GET THERE?

HOW ARE WE GOING TO GET THERE?

THE REQUIREMENTS

- Clicks on buttons and other interactives should run in the browser or on device
- Should be installable on a mobile device
- Should be runnable without internet
- Should have DataFlex DNA 



HOW ARE WE GOING TO GET THERE?

THE IDEA

- What we wanted to do was port the current DataFlex runtime
- That currently only works on Windows over to your local browser
- Allowing your local browser to run DataFlex code on device

HOW ARE WE GOING TO GET THERE?

THE IDEA

- Now, we had other ideas like creating a new JavaScript framework for example....
- Since....
- That seems to work for the other thousand companies
- But we want to keep our DataFlex DNA.

HOW ARE WE GOING TO GET THERE?

DATAFLEX DNA

Uniformity

One language to learn

Data-Binding controls

Entry_Item is all I need!

Out-of-the-box

One studio, click-and-run!

UI-driven objects

Unique object notation that is easy to understand

Business Rules

DataDicts ensuring validity and security

And everything else!

Simply the ease of DataFlex



TO GET THE WINDOWS RUNTIME
RUNNING IN THE BROWSER, JUST
HOW.....?

HOW ARE WE GOING TO GET THERE?

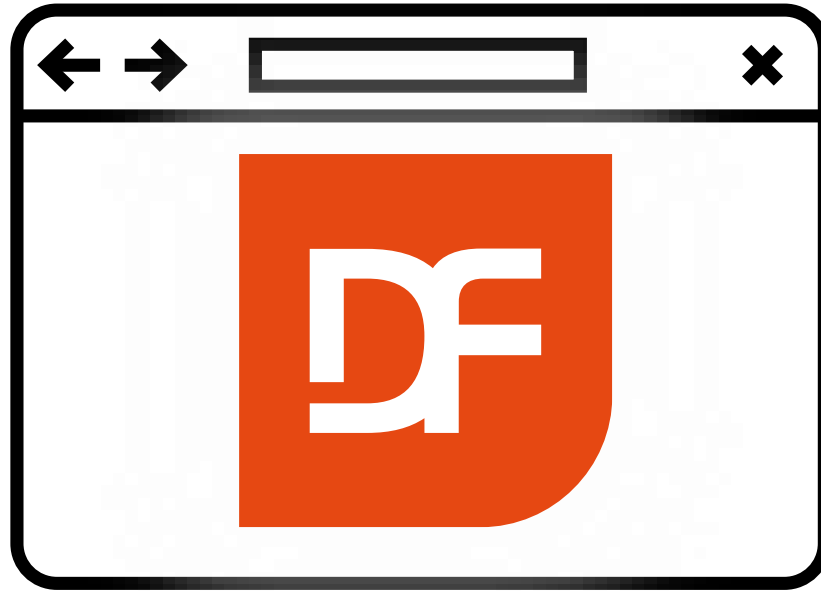
THE TECHNOLOGY

- WebAssembly is a technology from 2017 that is now coming into production environments
- It is an instruction format that runs in Web browsers
- Including FireFox, Chrome, Edge, Opera, Brave, Safari, ...
- The format is much smaller and pre-compiled allowing much more performance than running for example JavaScript
- It enables us to compile the DataFlex runtime to WebAssembly, allowing it to run DataFlex code in the browser



2017

HOW ARE WE GOING TO GET THERE?
THE TECHNOLOGY

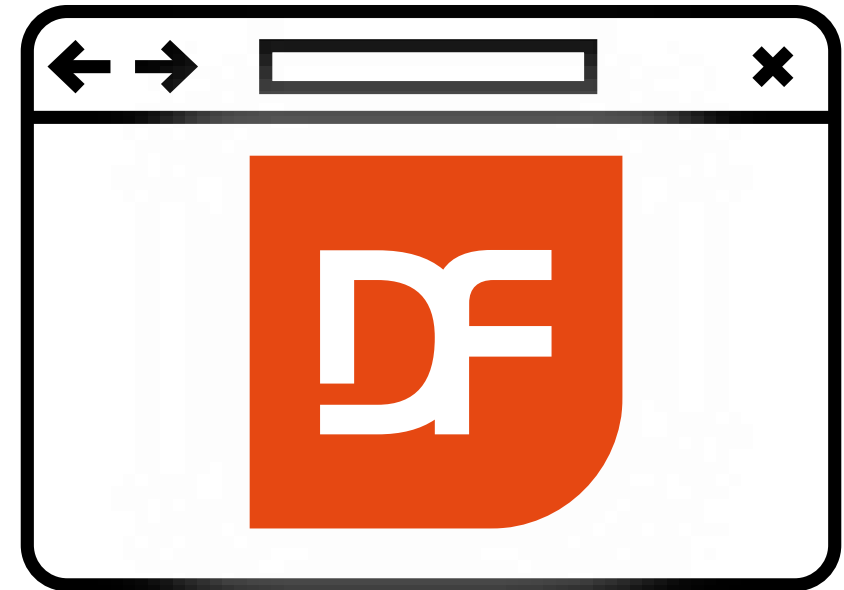


**A RUNTIME THAT WORKS IN
ANY BROWSER!**

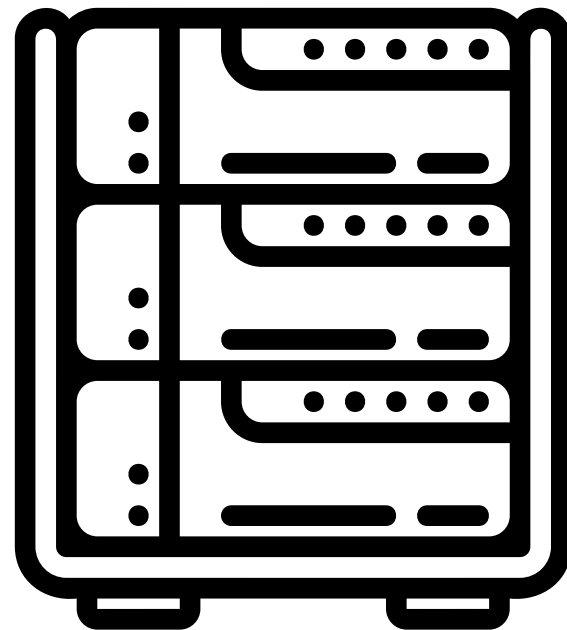
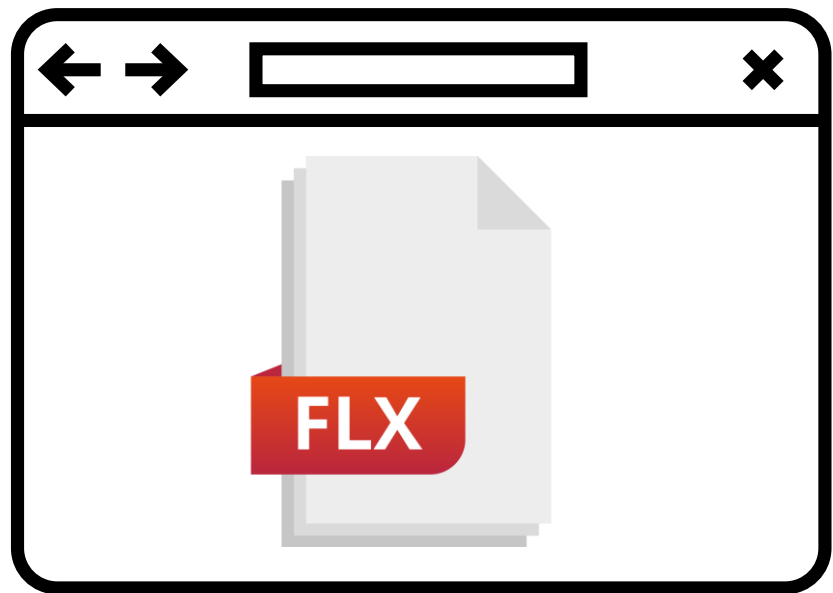
HOW ARE WE GOING TO GET THERE? THE TECHNOLOGY



Run Flex applications
in the browser

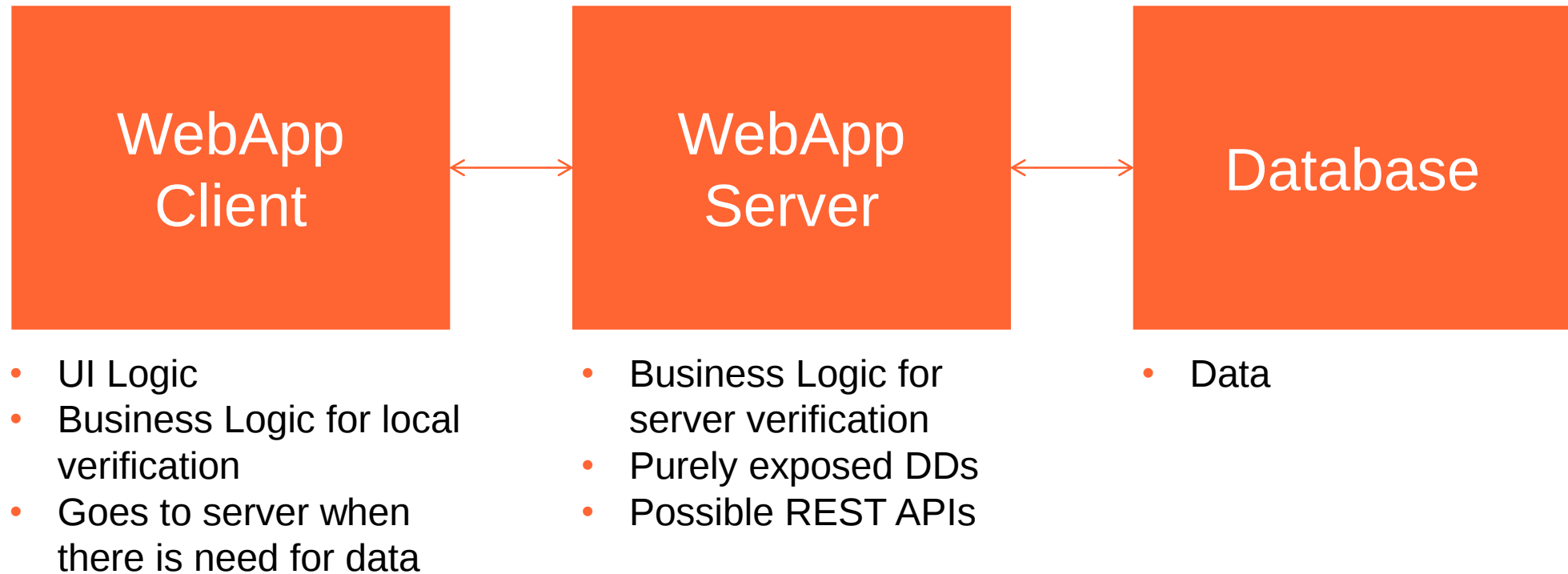


HOW ARE WE GOING TO GET THERE? THE TECHNOLOGY



HOW ARE WE GOING TO GET THERE?

THE TECHNOLOGY



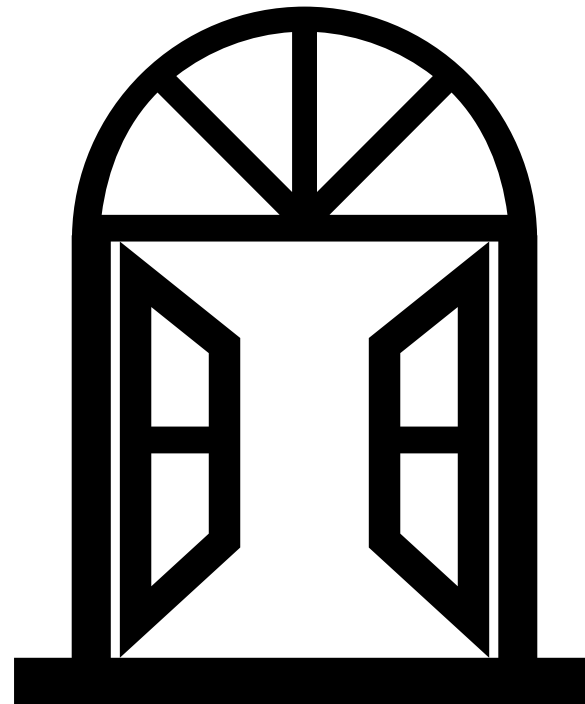
4

DATAFLEX TECH STACK

DATAFLEX TECH STACK – RUNTIME

NOT AS EASY AT WE'D HOPED

- In 1995, DataFlex became Windows only
- The good reason being that businesses (our focus) moved to Windows as well
- Non-Windows went mostly unmaintained for over almost 20 years
- Next to the fact that the complete codebase is over 40 already
- Porting this codebase to WebAssembly then went as well as you can imagine...
- Total:
- It would be a bad investment to try and reimplement newer code in such a codebase

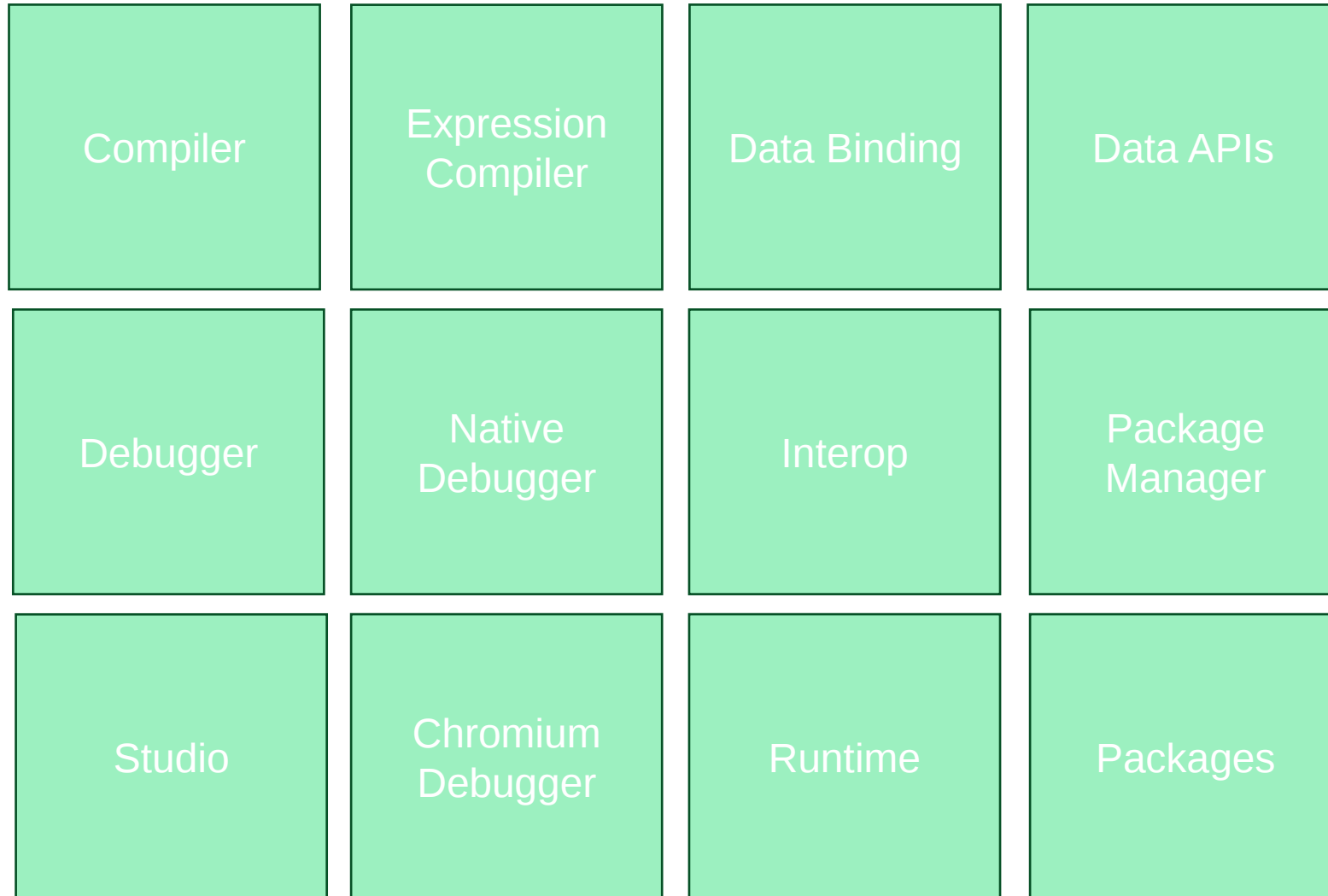


SO, WE DECIDED TO START FROM
SCRATCH 3 YEARS AGO!

Which is a big.... big.... project

DATAFLEX TECH STACK

IT TOUCHED EVERY PART OF THE ECOSYSTEM



BUILT FROM SCRATCH



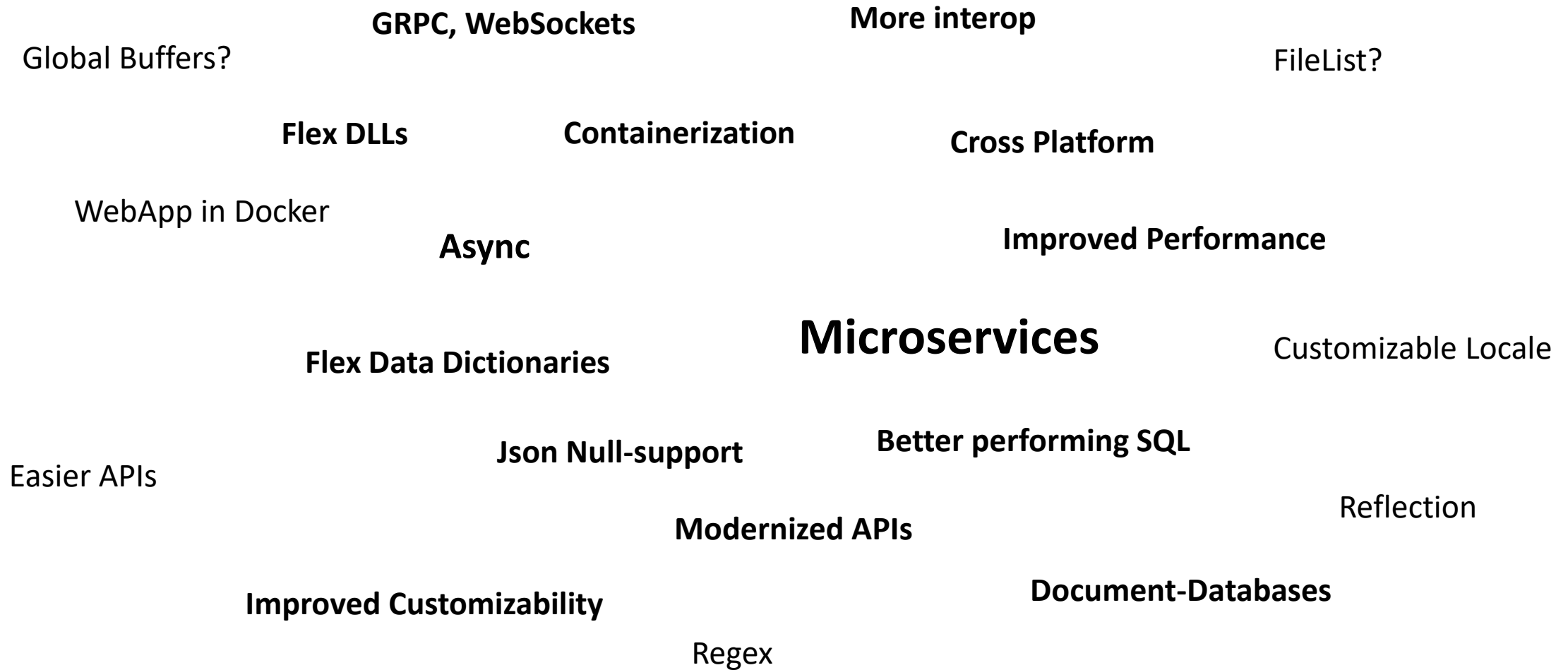
CHANGED OR UPDATED



NEW

NOW OBVIOUSLY THIS IS NO
LONGER ABOUT JUST MOBILE

DATAFLEX TECH STACK WISHLIST



DATAFLEX TECH STACK

WHAT ABOUT MY CODE?

- DataFlex remains DataFlex with its DNA
- There is no redesign of the language
- We intend to keep things as migratable as possible
- However, to be ready for the future some drastic changes are being experimented with to be ready for the next 30 years!



DATAFLEX TECH STACK

PORTABILITY

- Because DataFlex will once again run on all platforms (Windows, macOS, Linux, Unix, WebAssembly)
- APIs changed to become platform-agnostic
- For example, some classes and concepts have been rewritten:
 - cHttpTransfer → cHttpClient
 - cComAutomation → External_Class



5

EL PLAN, WHEN, AND WHAT?

BECAUSE OF THE SCOPE OF THE
PROJECT, WE SEPARATED
EVERYTHING INTO THREE PHASES

EL PLAN

THE THREE PHASES – PHASE 1

- Is focused on Web Applications, will provide offline support for WebApps
- Due to the WebAssembly Runtime; more responsive WebUIs are possible as everything is being processed on device
- Will allow for wrapping support on mobile, but is still to be determined in scope as to the automation of in-Studio functionality
- Provides both a new runtime for the Server (Windows) and Client (WebAssembly)

There will be a couple of different styles to work with DataFlex and it's new ecosystem:

- **TS Runtime on Server + TS Runtime on Client**

For the latest APIs and best integration for a full stack DataFlex application

- **TS Runtime on Client + Any (REST)**

For the best WebApp UI framework on the client but with a different backend

- **TS Runtime on Server (previous paradigm)**

For the latest in DataFlex technology but using the server-only paradigm for simplicity

- **Cur Runtime on Server (previous paradigm)**

If it works, don't touch it! Will be supported for a long while

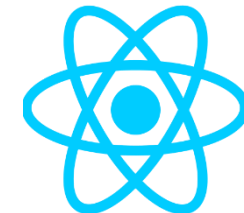
- **Cur Runtime on Server + TS Runtime on Client**

Only supported using REST

EL PLAN

THE THREE PHASES – PHASE 1

- This phase allows businesses to compete even more with other fat client frameworks like React, Angular, Svelt, and others on all levels
- But still with the ease of DataFlex, now with more customizability



- Will add built-in cross-platform support for DataFlex applications including:
 - **New WebApp server for all (other) platforms**
Mimics hosting models closer to other languages with custom load-balancing, etc.
 - **Containerization support in Kubernetes/Docker clouds**
With pre-built dataflex-runtime/-server images!

Which will allow for;

- More scalability of your applications
- More advanced hosting options in different environments than Windows –IIS, making hosting a lot cheaper in a lot of cases

EL PLAN

THE THREE PHASES – PHASE 3

Mostly unplanned but topics we are looking into are:

- Become general purpose again (as up until phase 2 is web app focussed)
- Add (new) technologies for using DataFlex as a local application on all platforms

And, of course wishlist items

EL PLAN WHEN?

- First we will have one more release (25.0) with amazing studio tooling and other features
- After that the Tech Stack phases are planned to be released over time, but is the main focus!



EL PLAN

WHAT ARE WE WAITING FOR?

- A whole lot works, but...
- We are working on reworked Data APIs that take a lot of careful deliberation
- A lot of Studio work is left to do
- And like I said, we still have to look into Mobile App-stuff to look into



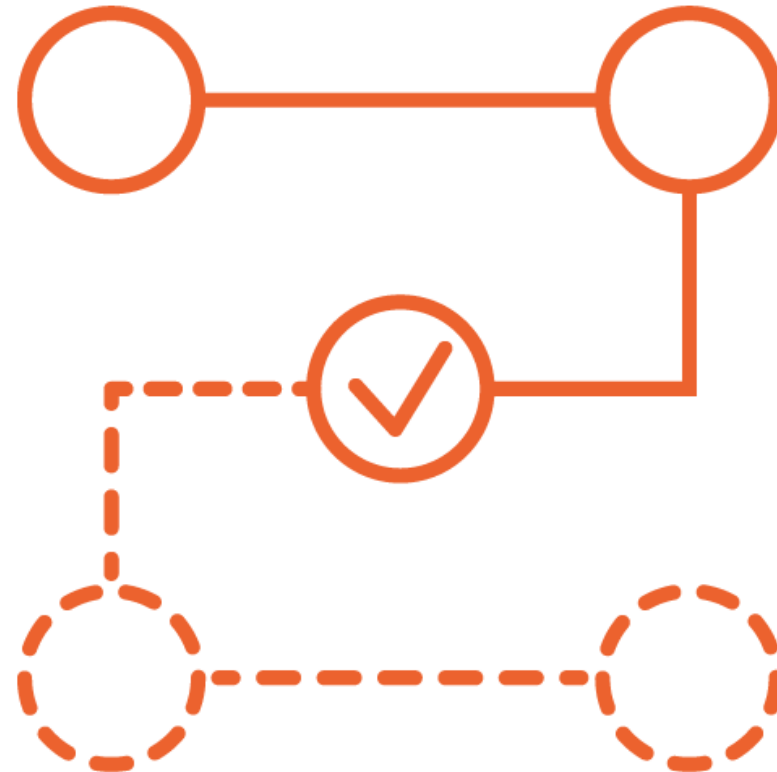
BUT...

WE HOPE TO BRING YOU A
TECHNICAL PREVIEW SOON AFTER
25.0

EL PLAN

WHAT ABOUT THE CURRENT RUNTIME?

- The new runtime will slowly take over platforms in multiple releases
- But even after the tech stack phases we will support the current runtime for a considerable time like we have done up till now
- With features, bug fixes, and patches included with some backports from TS already!



ALL IN ALL, THE FUTURE LOOKS
BRIGHT!

AND THINK IT WILL BRING YOUR
APPLICATIONS BEYOND HORIZONS!