DataFlex

DATAFLEX TECHNOLOGY STACK PROJECT

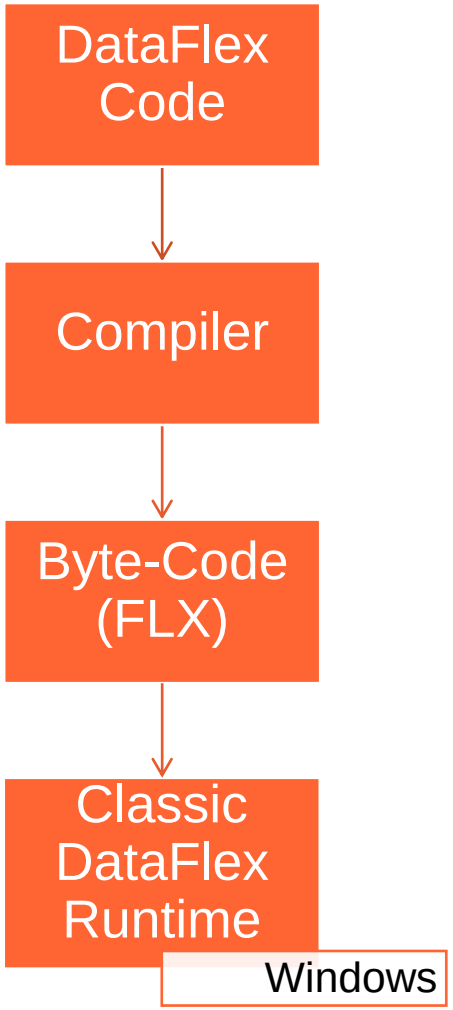
HARM WIBIER

SO, WE ARE A BUILDING A NEW DATAFLEX RUNTIME..

INTRODUCTION
WHAT IS A RUNTIME?

- The compiler converts DataFlex code into binary instructions (byte code)
- These instructions are stored a binary file called the FLX together with data like constants
- On windows this FLX is inserted into an .exe file
- This .exe file loads the runtime (vdfvm.dll) and calls it with a pointer to the FLX

- Java and .NET work similarly

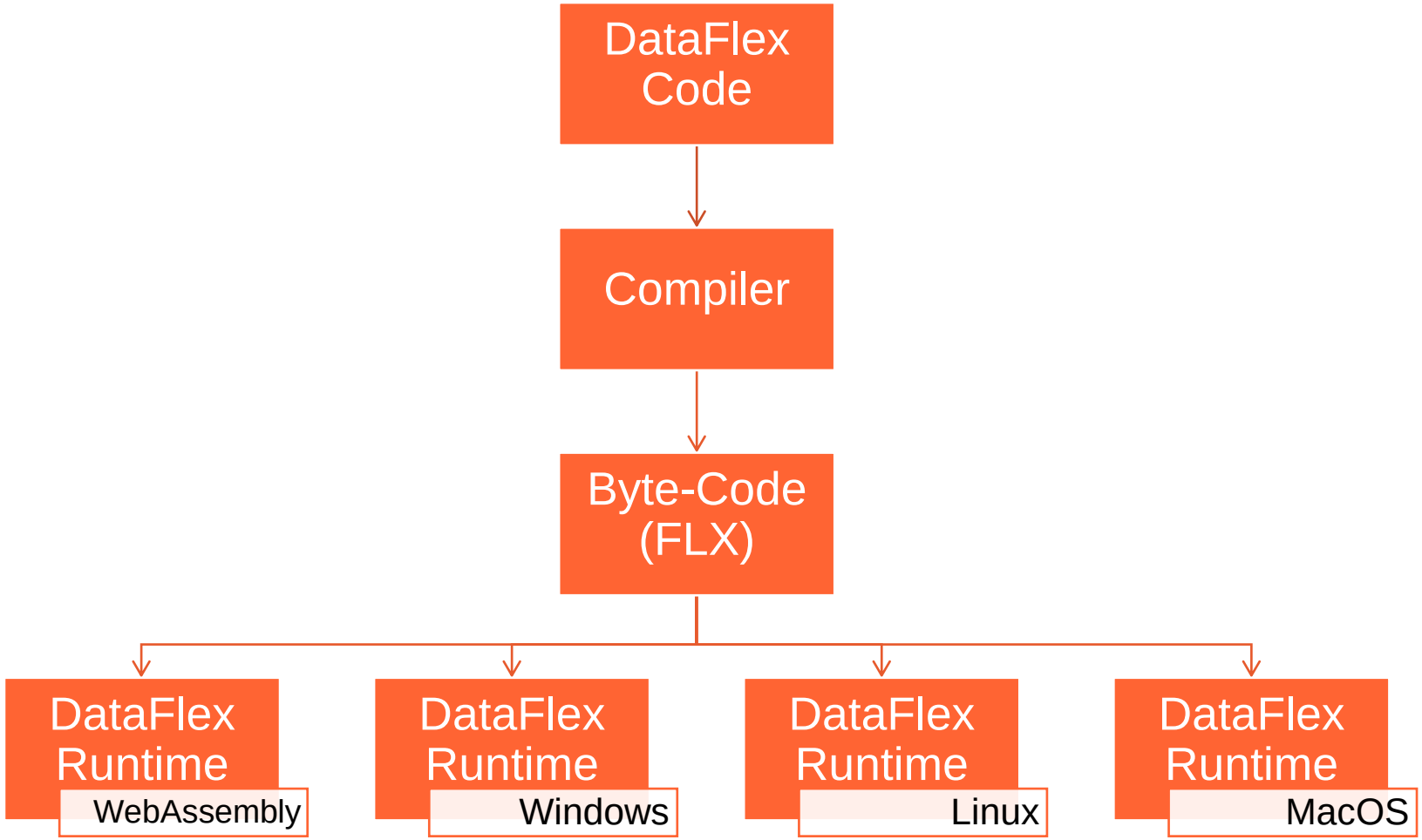

DataFlex
Code
Compiler
Byte-Code
(FLX)
Classic
DataFlex
Runtime
Windows

INTRODUCTION
THIS NEW RUNTIME..

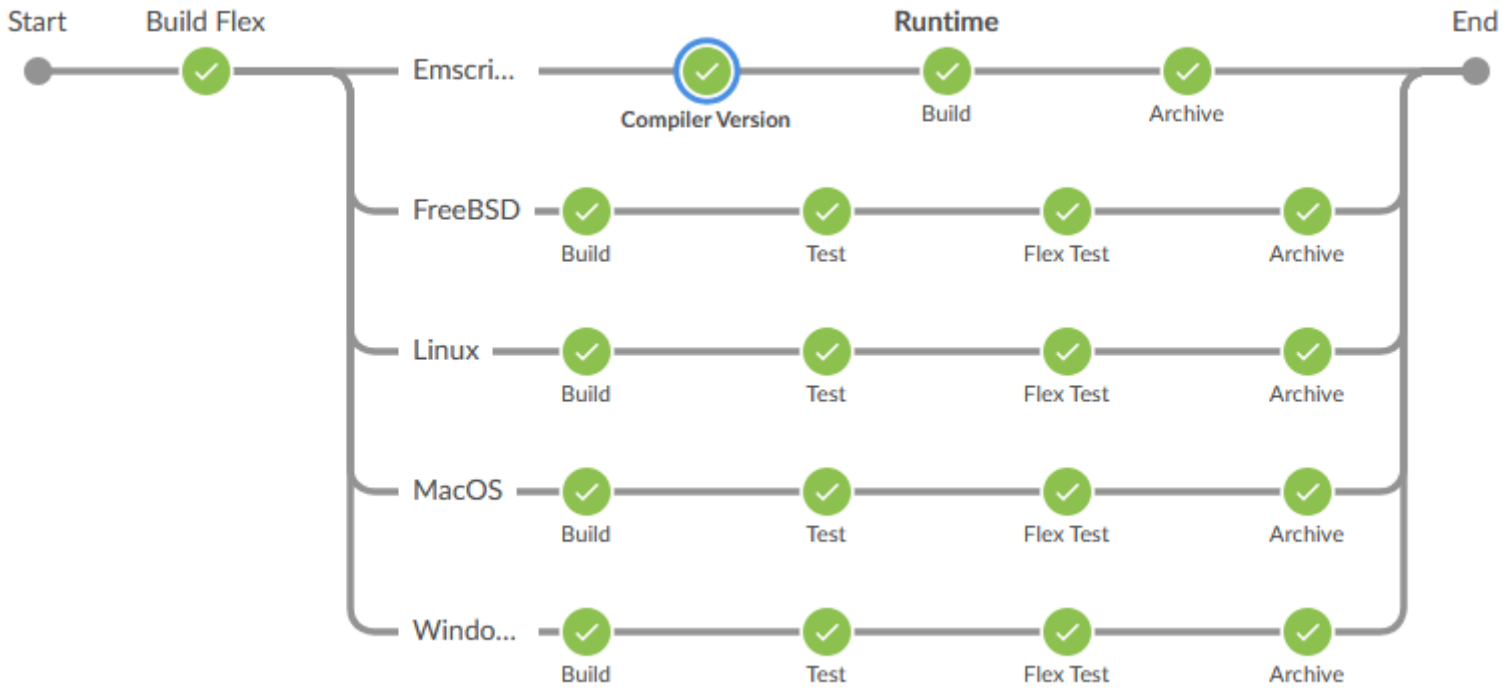
- Clean modern C++ codebase
- Is going to run on multiple platforms
  - WebAssembly
  - Windows
  - Linux
  - MacOS
  - …
- Compatible with the existing FLX format & compiler

INTRODUCTION
WHAT WILL IT BRING US..

- Maintainability
  - Clean codebase
  - Knowledge
- Lean
  - Incorporate new technologies better
- New platforms
  - Developed with platform independency in mind


DataFlex Code
Compiler
Byte-Code (FLX)
DataFlex Runtime
WebAssembly
DataFlex Runtime
Windows
DataFlex Runtime
Linux
DataFlex Runtime
MacOS

INTRODUCTION
PLATFORMS THE RUNTIME IS CURRENTLY TESTED ON



01

WEBASSEMBLY

WEBASSEMBLY
THE FIRST NEW PLATFORM

- WebAssembly (WASM) defined a portable binary-code format for executable programs with the goal to facilitate high-performance applications on web pages

- Run ‘native’ code inside the browser
- Compile C / C++ code to WebAssembly using Emscripten

WEBASSEMBLY
WHAT CAN YOU DO WITH DATAFLEX IN THE BROWSER?

- Talk and respond directly to the Web Controls in JavaScript
  - Offline: Without going to the server!
- Talk to the server
  - As most applications will require a central server for sharing data
- Work with offline storage

WEBASSEMBLY
IN PRACTICE

- dfvm.wasm is the WebAssembly version of our new runtime!
- It runs the flx (which in this case it downloads from the server)
- Debugging using the Chromium DevTools Protocol (CDP)
- Runs inside its own worker
  - We do not want to block the browser its main thread

02

WINDOWS

WINDOWS
WEBAPPS ON WINDOWS

- Something needs to run on the server to provide the data
- In the future we also want this to be Linux (inside a docker image)
  - *Which means a new WebApp Server without IIS dependency*
- But as a first step it was easier to use to the existing WebApp Server
- But for the new database logic we couldn’t use the existing runtime (soon to be called Classic runtime)

WINDOWS
RUNNING WEBAPPS WITH THE NEW RUNTIME

- New windows debugger
- Added COM support (to communicate with WebApp Server)

03

DATA BINDING

DATA BINDING
DATAFLEX IS ALL ABOUT RELATIONAL DATABASES

- Database API
  - Find / Save / Delete records using commands
  - Relate finds parent records
- Multiple drivers for different databases
- Data Dictionaries
  - Business rules
  - Data binding

DATA BINDING CHALLENGES

- DataFlex front-end in the browser that does not have direct database access
  - Needs to do data binding
  - Needs to perform validations
  - Needs to know the data model
- DataFlex back-end on the server that exposes an HTTP API for accessing data
  - Needs to perform validations
  - Needs to know the data model
  - Needs to access the database

- This is where DataFlex should make the difference!

DATA BINDING
GOALS

- Define the data model & business rules once
  - And use them on the client and the server
- Provide data binding on the client
  - Design user interfaces and data structures like we do for windows or existing webapps while data goes over the line
- Open up data connectivitity
  - Customize logic between Data Dictionaries and driver
- Optimized data loading for sets of data
  - Single API call to fill a (Web / CJ) List
- Improve debuggability of Data Dictionaries
- Get rid of confusion between global and local file buffer

DATA BINDING
THE IDEA'S

- Compile database model into DataFlex applications
  - Global entity definitions
  - Use structs as file buffers
- Make Database API Stateless
  - Define a message format with requests / responses for database operations
- Remove global file buffers and redirect commands to local file buffers
- Rewrite Data Dictionaries in Flex code
  - That talk to a data adapter that might be written in Flex Code

DESIGNING DATA BINDING…
ENTITY DEFINITION

```
Entity Customer
    { PrimaryKey=On }
    { AutoIncrement=On }
    { Size=8 }
    Integer Number
    { Size=30 }
    String Name
    { Size=30 }
    String Address
    { Size=30 }
    String City
    { Size=2 }
    String State
    { Size=100 }
    String Zip
    { Size=8.2 }
    Number Balance
    Boolean Status
End_Entity
```

04

OVERVIEW

TECHSTACK I

- New runtime for WebAssembly
  - Implement faster WebApps that only communicate the data once they are loaded
  - Implement offline Apps
    - Progressive WebApps
    - Wrap using PhoneGap / Cordova like technology
- New runtime for Windows
  - Implement the server-side API’s for new WebApps
- New data binding

TECHSTACK II

- **TechStack II**
  - Host WebApps on multiple platforms
    - Linux
    - Docker
  - New WebApp Server
- **TechStack III**
  - Desktop Apps on multiple platforms
  - FlexTron?

PLANNING…
WHEN?

- Release ‘TechStack I’ after DataFlex 2025
- Linked to the package manager with its Toolchain Packages
- With package manager we can do automatic updates
  - Respond quickly to bugs
  - Add missing features

- Hoping to do a technology preview soon..

DataFlex

THANK YOU FOR YOUR TIME!