



Integration in a serverless world

Mohammed Brückner @ platformeconomics.com

Mohammed Brückner // "All views are my own"



Prefix: A difficult relationship

An exhausting tale of IT "value" delivery

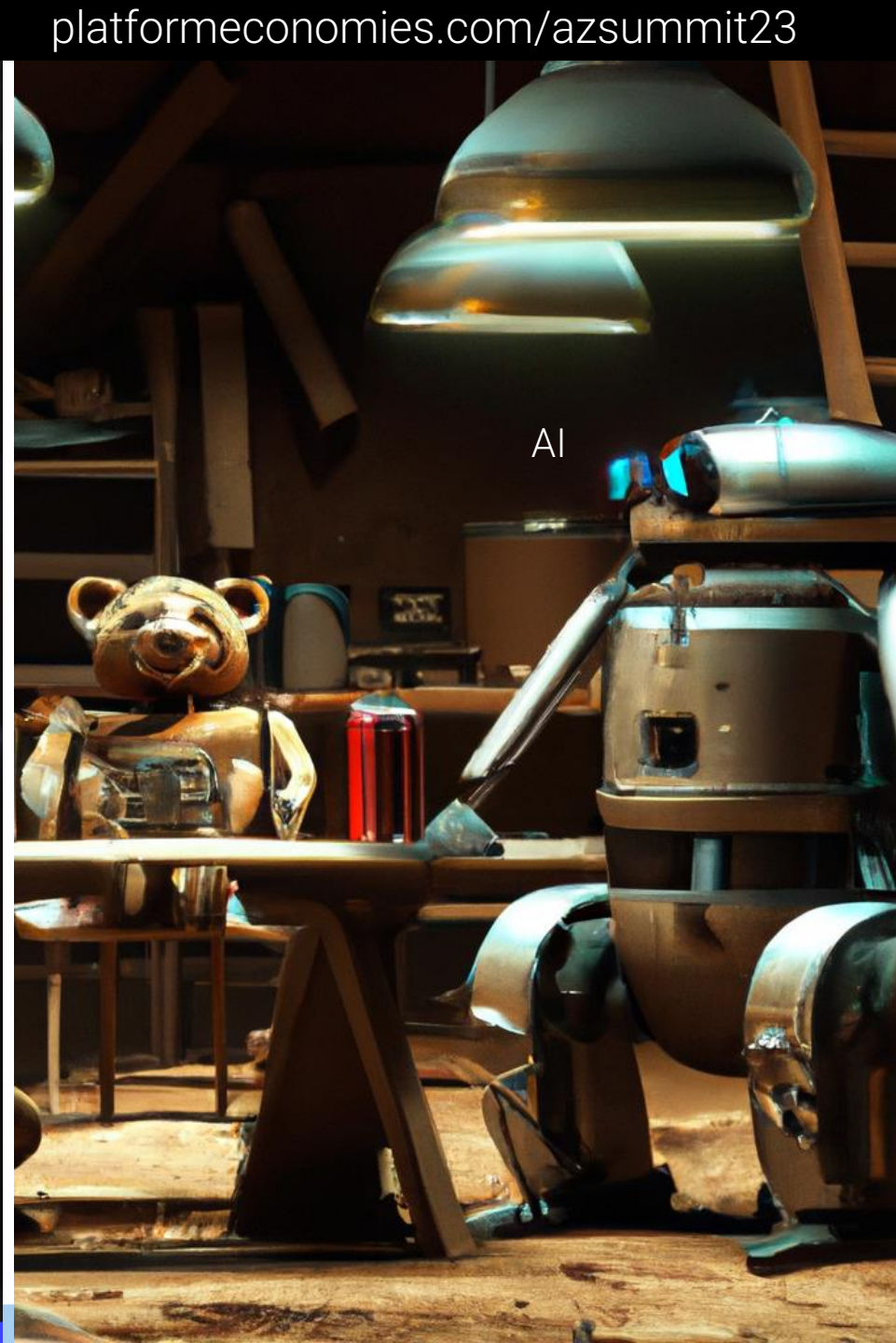




“Business” Stakeholder

IT “Delivery Partner”

“Project Manager”



AI





“Business” Stakeholder

“Requirements”



Frustrated Developers



Frustrated CIO



GOODBYE, WATERFALL!

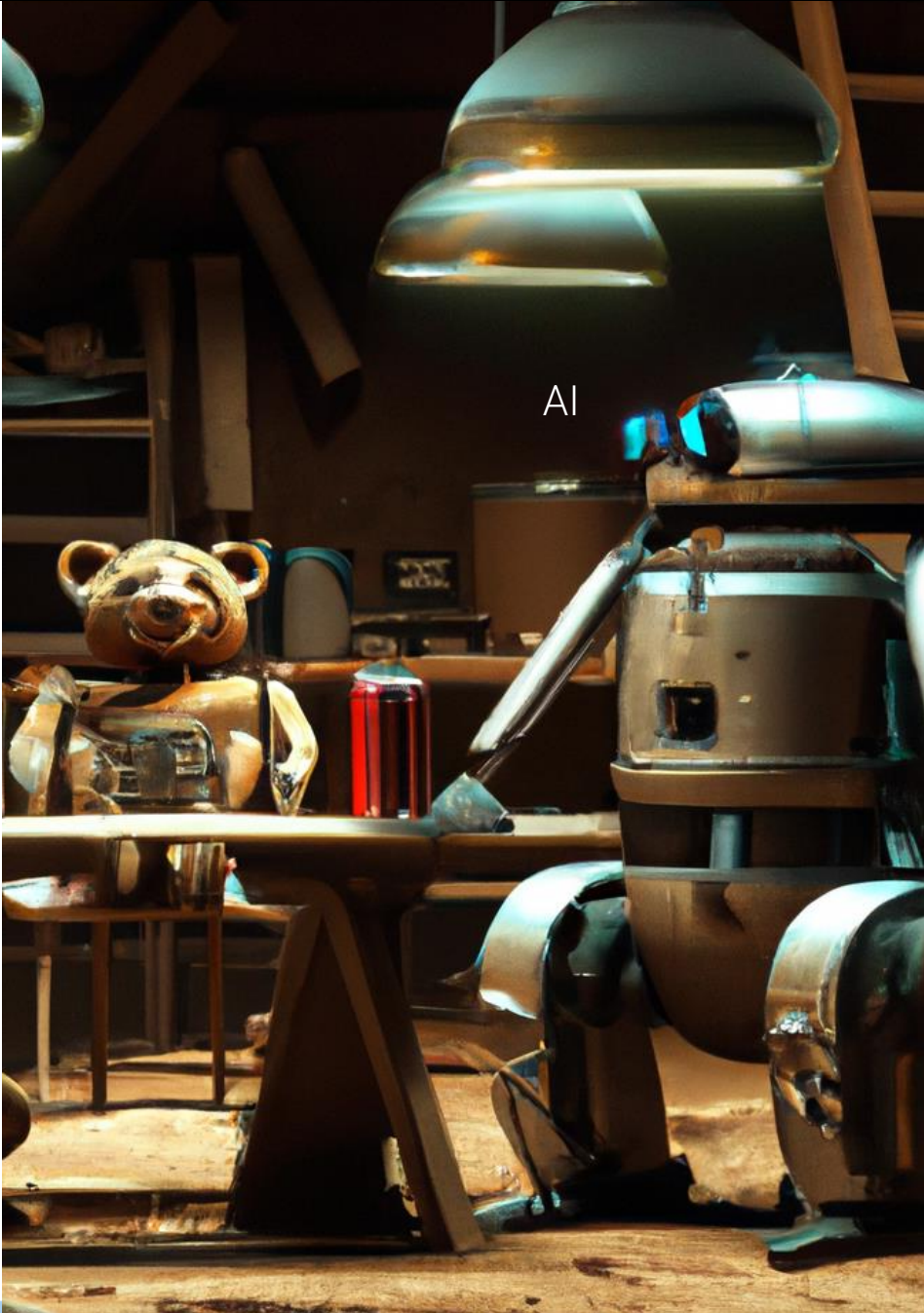
WELCOME, AGILE!



"Business" Stakeholder

"IT Product Owner"

"Scrum Master"



AI



“Business” Stakeholder

“IT Product Owner”

“Scrum Master”

Closer to the Business with a chance to “participate” (“seat at the table”)

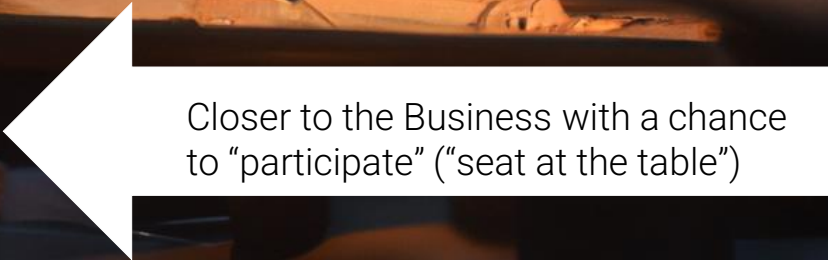


How is Business participating in the creation of IT in the other direction? (They aren't.)

“Business” Stakeholder

“IT Product Owner”

“Scrum Master”



Closer to the Business with a chance to “participate” (“seat at the table”)

WELCOME, LOW-CODE!



BEST RUN COMPANIES RUN ON EXCEL



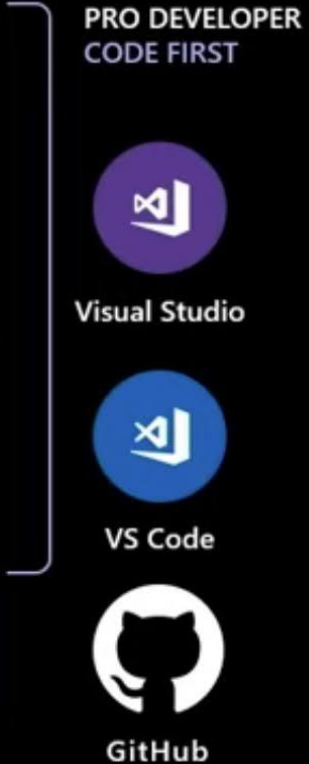
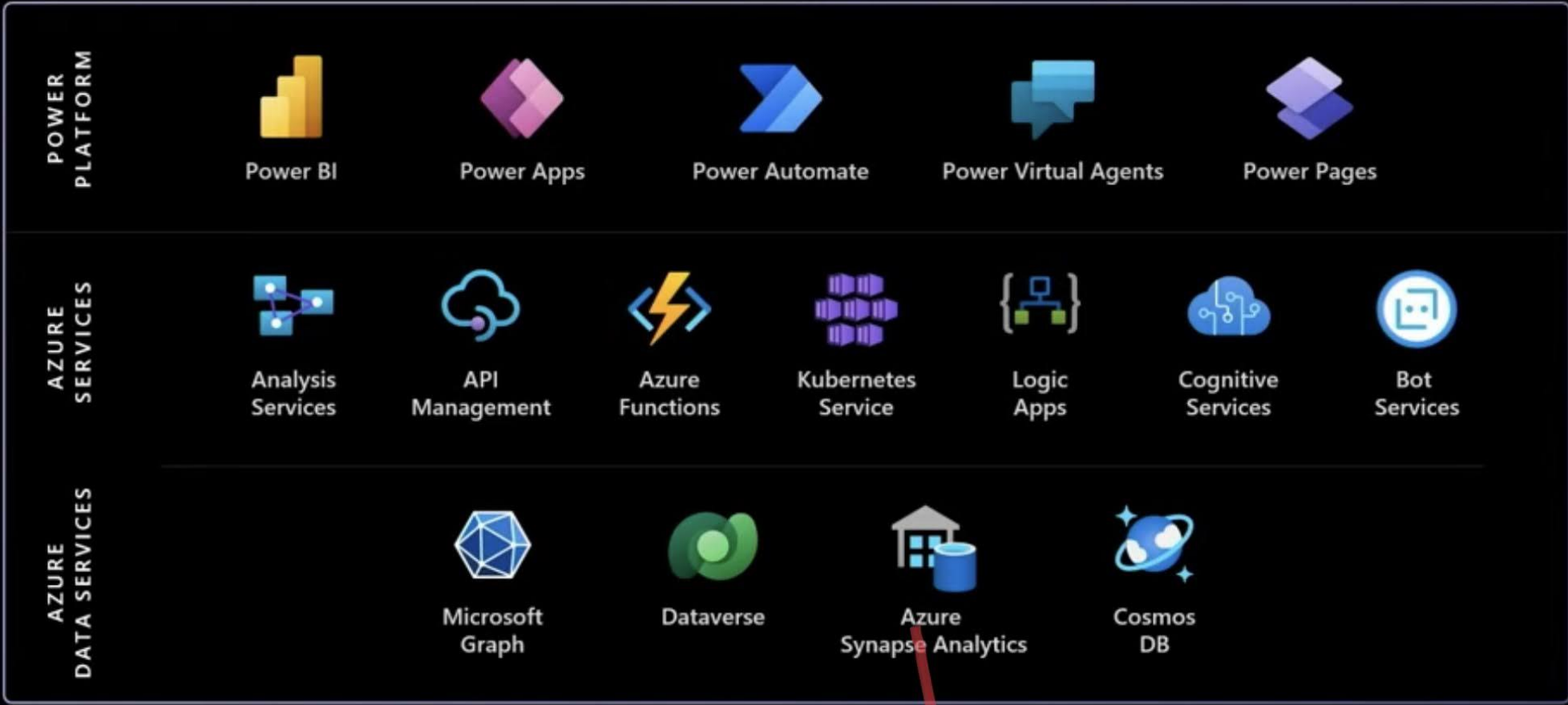
What are benefits of combining low-code and pro-code?

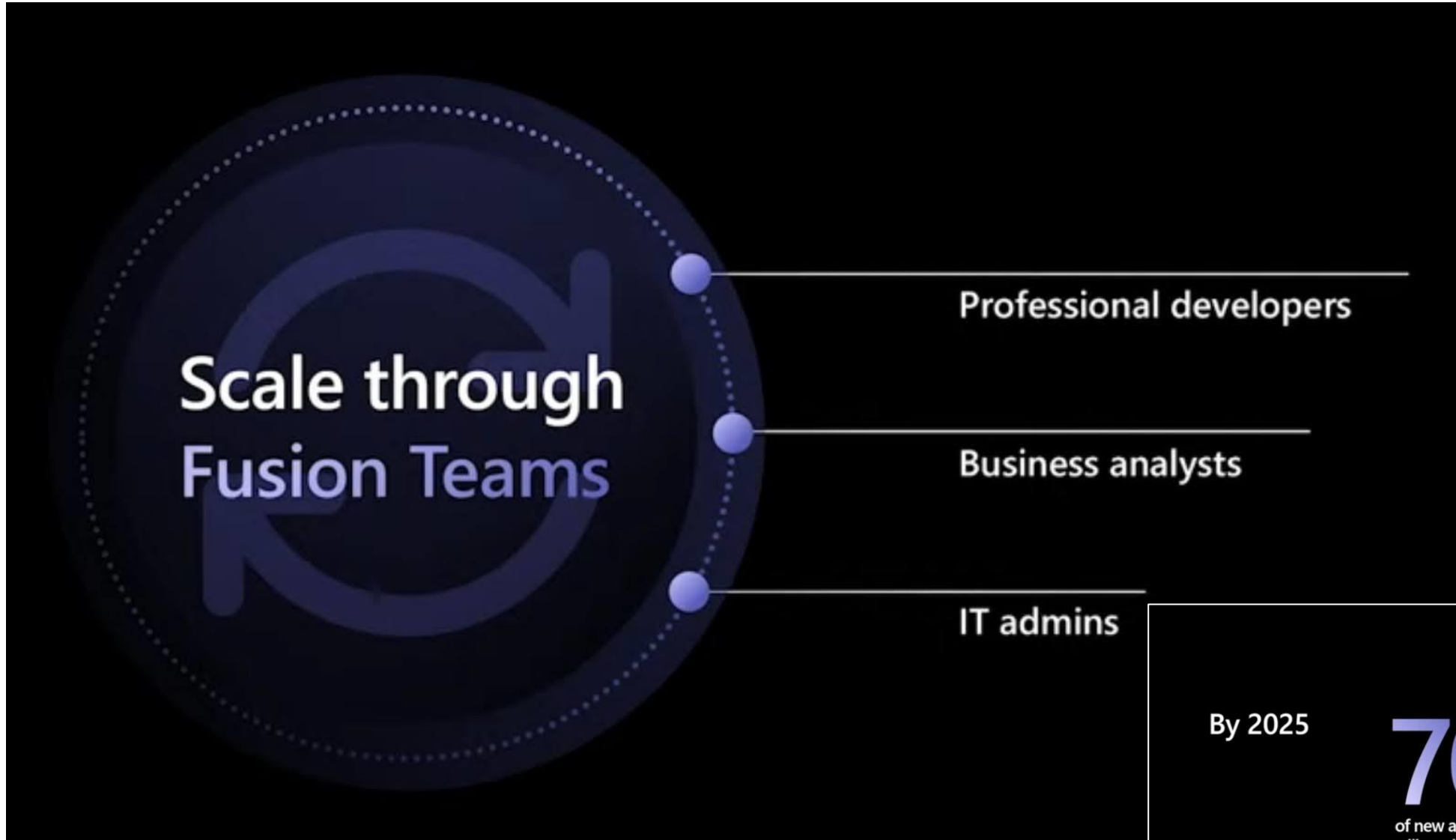
Some benefits of combining low-code and pro-code include:

- **Faster development:** Low-code platforms can be used to quickly prototype and build the basic structure of an application, while pro-code can be used to add custom functionality and fine-tune the application.
- **Improved collaboration:** Low-code platforms can be used by non-technical team members, such as business analysts, to contribute to the development process. This can improve collaboration and communication within the team.^{[0][2]}
- **Increased flexibility:** Combining low-code and pro-code allows for a balance between rapid development and customizability. This can be useful in situations where the requirements of the project are not fully known or are subject to change.^[0]
- **Better performance:** Pro-code can be used to optimize the performance of an application, which can be especially important for applications with high traffic or complex data processing requirements.

Overall, combining low-code and pro-code can provide a more efficient and flexible approach to application development.

Pro Code + Low-Code = No Limits





By 2025

70%

of new applications developed by enterprises will use low-code or no-code technologies

COMPARE: [#LOWCODEFEBRUARY](#)

Microsoft ist „Leader“ im „Magic Quadrant for Enterprise Integration as a Service 2023“

5 Reasons Enterprises choose Azure Integration Services

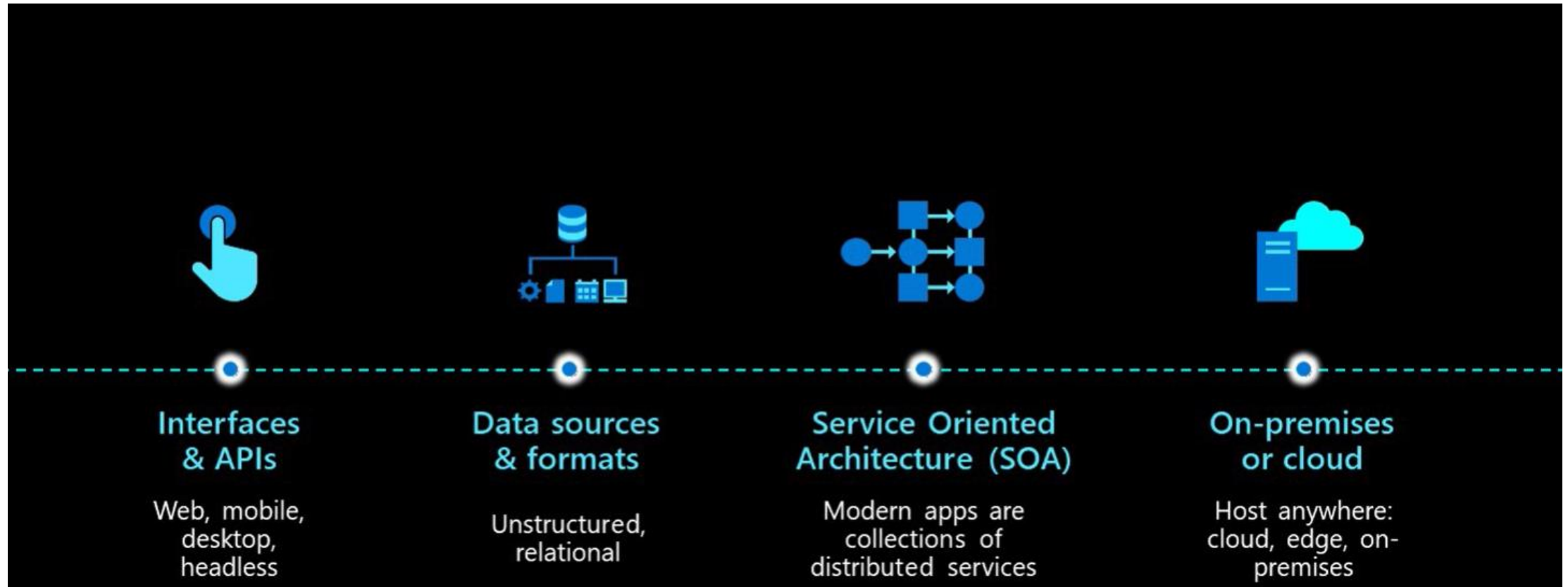
Continuous innovation

Globally managed offerings

Strong security and compliance

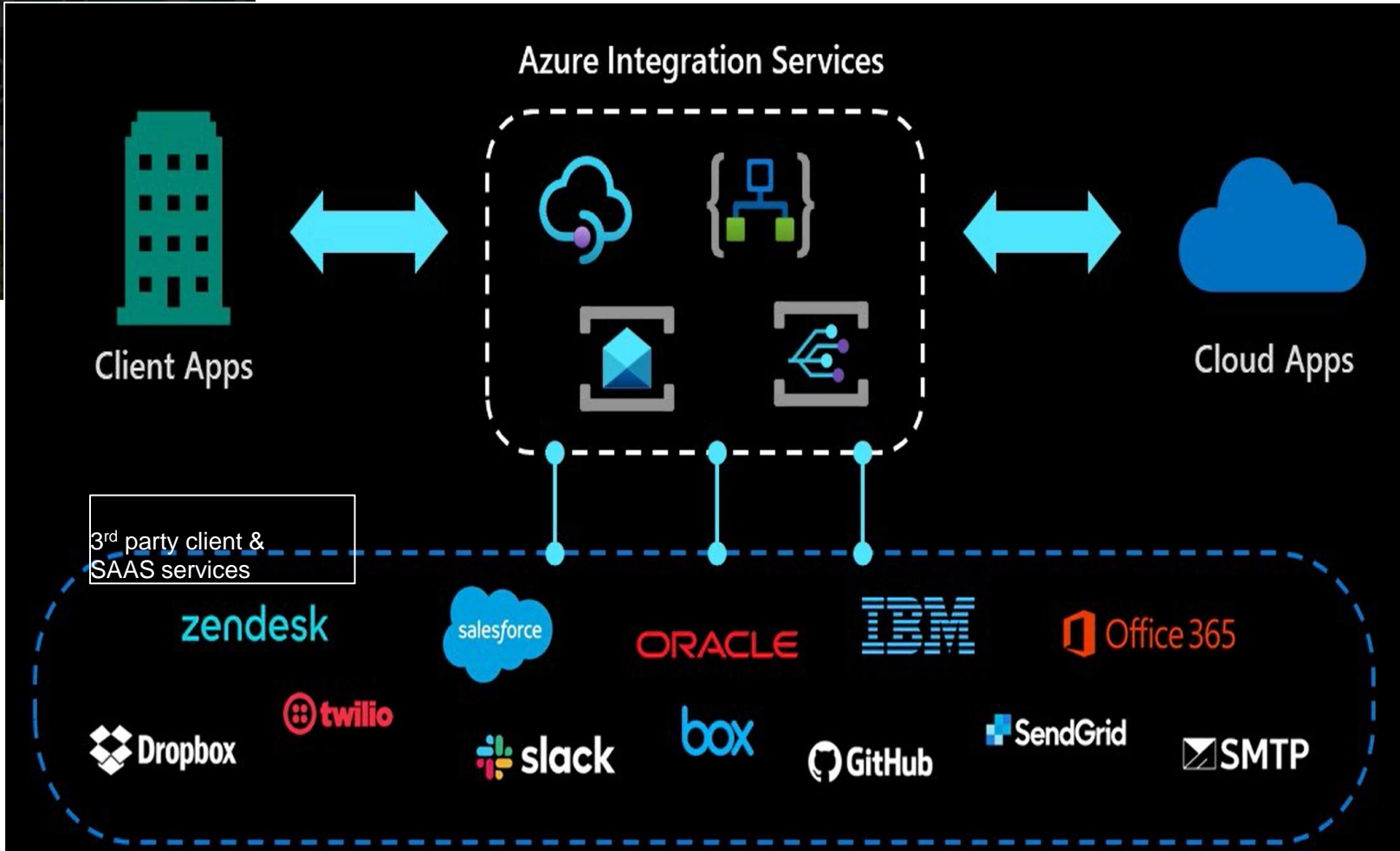
Flexible pricing

Supports both citizen and professional developers



COMPARE: [#LOWCODEFEBRUARY](#)

900+
Data connectors

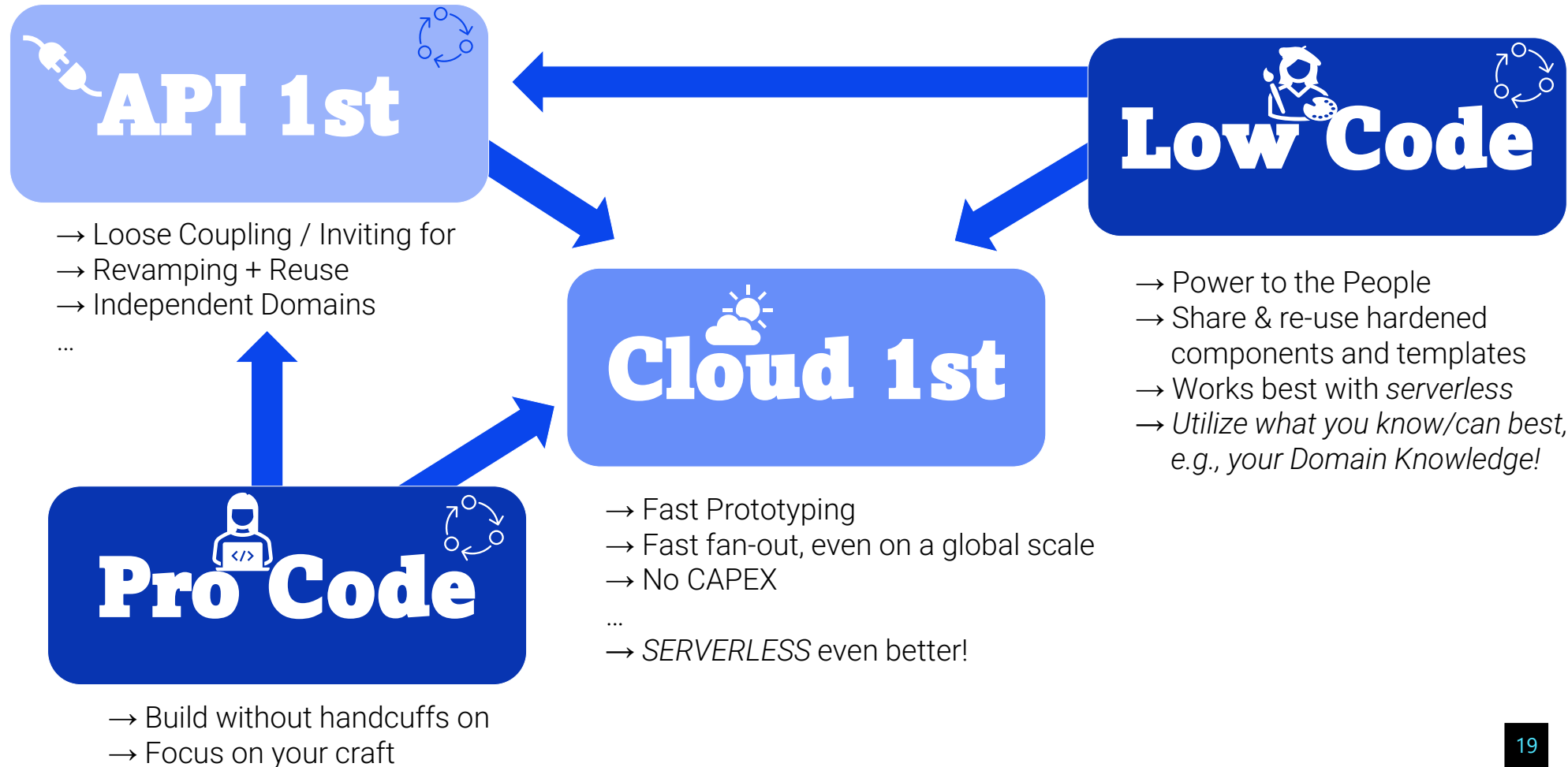


COMPARE: [#LOWCODEFEBRUARY](#)

The "Secret Sauce" for success of Fusion Teams

Conviction

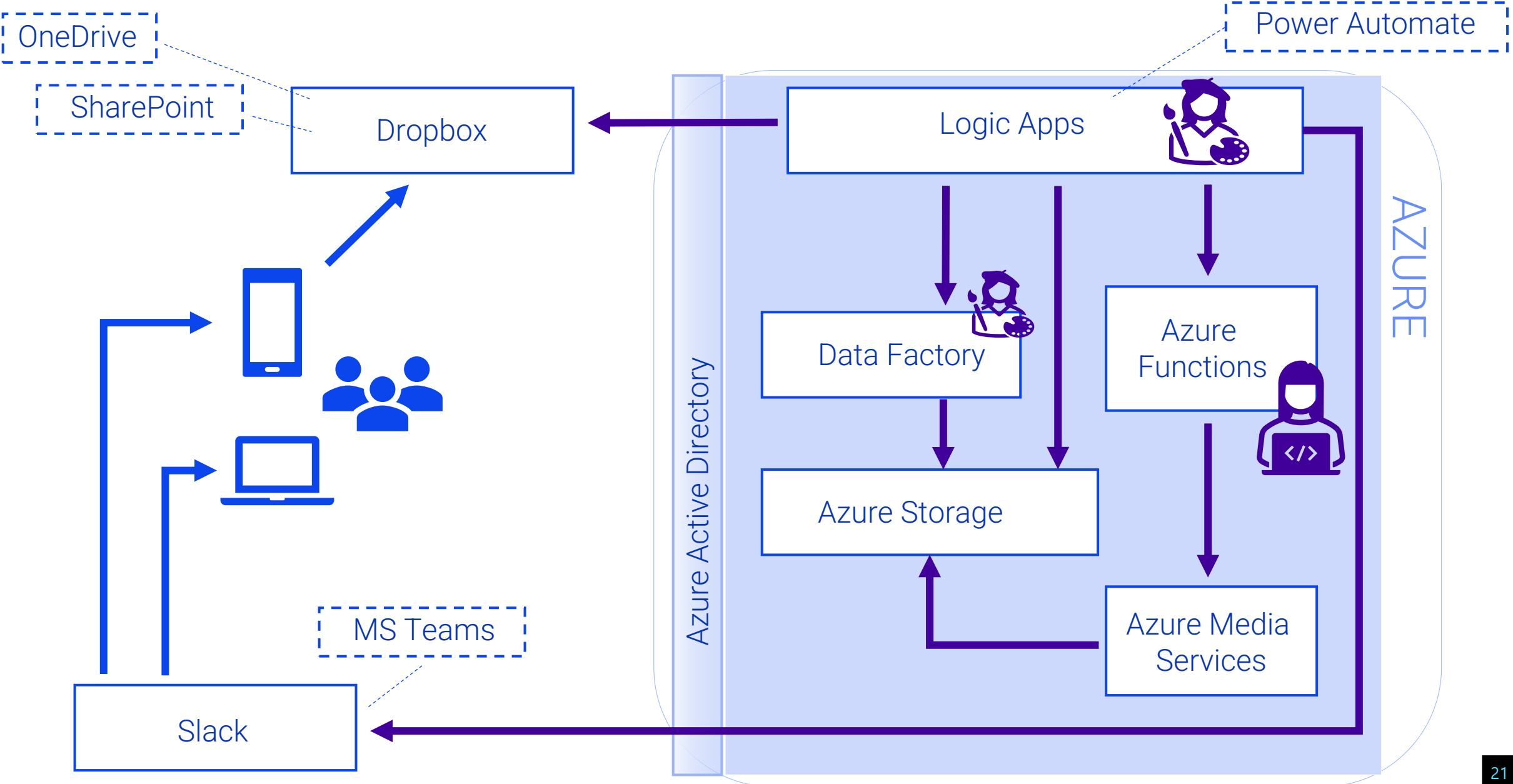
The Operating Model needs to evolve from Centralized "Silo Teams" to **autonomous, Interdisciplinary, purpose driven** Teams. Focussing on **Service Domains** supporting **continuous product life cycles** – instead of time bound project life cycles.



And now, to the fun part

EXAMPLE SCENARIO
WALK-THROUGH



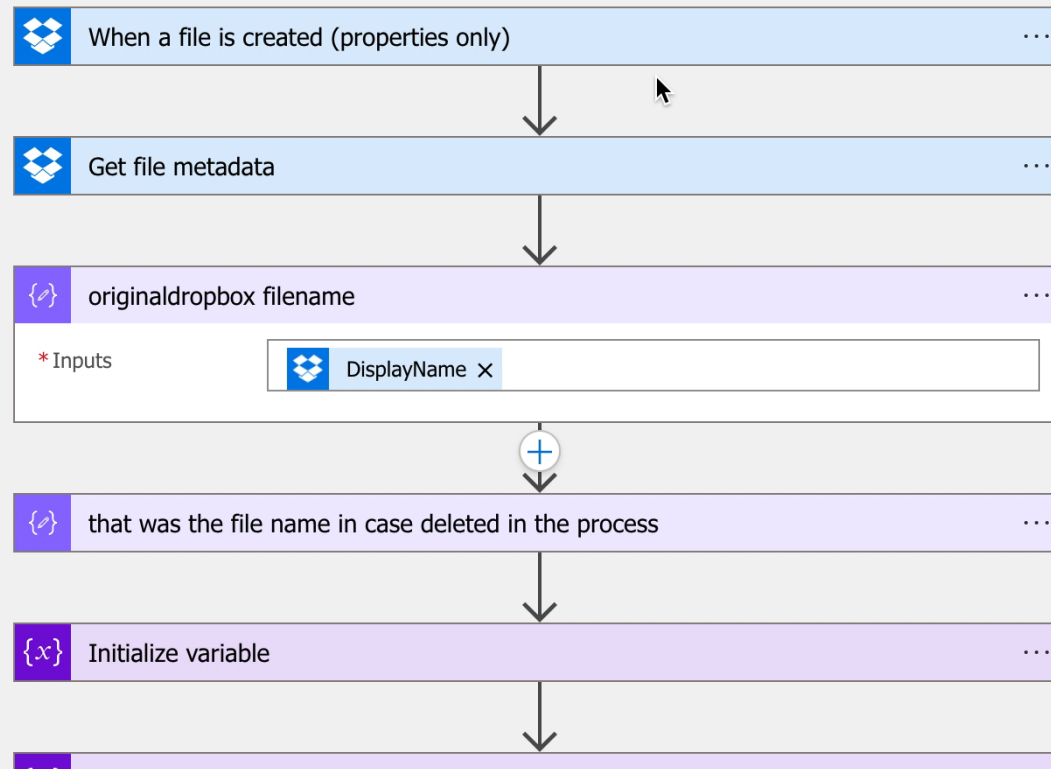


Logic Apps Designer



Save Discard Run Trigger Designer Code view Parameters Templates Connectors Help Info

100%



obtain a valid access token

Parse JSON



Create a pipeline run

DataFactory will use application/octet-stream as content-type when copying over files to BlobStorage, which means images etc will not shown in browser but downloaded -as of 2022 it seems there is no way to set the content-type

* Subscription [Redacted]

* Resource Group VSE_df

* Data Factory Name vsemodfactory

* Data Factory Pipeline Name pipeline1

```
parameters
{
  "a_token": "{ access_token }",
  "container": "pubtmp",
  "file_name": "{ DisplayName }",
  "modified_file_name": "{ Outputs }",
  "subfolder": "large4publish"
}
```

Add new parameter



do plenty of stuff to convert the video file

{x} Set variable 2



⚡ kickofftransformationams

Request Body

```
{  
  "filesasuri": "Web Url x",  
  "origfilename": "Outputs x"  
}
```

Add new parameter



{ } Parse JSON 2

Data Factory | Validate all | Publish all | Preview experience Off

Factory Resources

- Pipelines (4)
 - copyfilefromblob2fileshare
 - copyfilefromBlobContainer2BlobCon...
 - copyfilefromPub2Pub
 - pipeline1**
 - Change data capture (preview) 0
 - Datasets 5
 - Data flows 0
 - Power Query 0

Activities

- Move & transform
- Synapse
- Azure Data Explorer
- Azure Function
- Batch Service
- Databricks
- Data Lake Analytics
- General
- HDInsight
- Iteration & conditionals
- Machine Learning
- Power Query

pipeline1

Validate | Validate copy runtime | Debug | Add trigger

Copy data

get data from dropbox

General | **Source** | Sink | Mapping | Settings | User properties

Source dataset * | dropdown: dropboxdata | Open | New | Learn more

Request method * | dropdown: POST

Additional headers | dropdown: Authorization: Bearer @{pipeline().par...



index.ts — vsstudiogetgroovy

EXPLORER

- GEÖFFNETE EDITOREN
- VSSTUDIOGETGROOVY
 - .vscode
 - azure-functions-core-tools
 - blobshuffler
 - deleteteamsassets
 - dist
 - kickofftransformationams
 - blobStorage.ts U
 - encodingJobHelpers.ts U
 - function.json U
 - heavyduty.ts U
 - index.ts U**
 - sample.dat U
 - transformFactory.ts U
 - node_modules
 - .funcignore U
 - .gitignore U
 - host.json U
 - local.settings.json U
 - package-lock.json U
 - package.json U
 - tsconfig.json U

TS index.ts U X

kickofftransformationams > TS index.ts > [⌘] httpTrigger


```

1  import {
2      AzureMediaServices, KnownAacAudioProfile, KnownH264Complexity,
3      StandardEncoderPreset, AssetContainerPermission
4  } from '@azure/arm-mediaservices';
5  import { AzureFunction, Context, HttpRequest } from "@azure/functions";
6  import { ManagedIdentityCredential } from "@azure/identity";
7  import { v4 as uuidv4 } from 'uuid';
8  import * as jobHelper from "./encodingJobHelpers"; //a typescript (class) file named like
9  import * as factory from "./transformFactory"; //a typescript (class) file named like thi
10 import * as url from 'whatwg-url';
11
12 // This is the main Media Services client object
13 let mediaServicesClient: AzureMediaServices;
14 //Talking Azure Subscription
15 const subscriptionId: string = ████████████████████████████████████████████████████████████
16 //RG in which the Azure Media Services Account resides
17 const resourceGroup: string = "vsmediaservcs";
18 //Azure Media Services Account
19 const accountName: string = "vsemom mediaservcs";
20 let inputFile: string;
21 // This is SAS signed URL, valid only for a determined time at creation
22 //let inputUrl: string = "https://propmsmaterial.blob.core.windows.net/pubtmp/schneekram
23 let inputUrl: string; ████████████████████████████████████████████████████████████
  
```

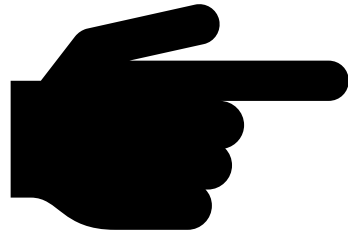
PROBLEME | TERMINAL | AZURE | AUSGABE | DEBUGGING-KONSOLE

zsh + - [] [X] [^] [X]

moes@Moooorrrs-MacBook-Pro vsstudiogetgroovy %



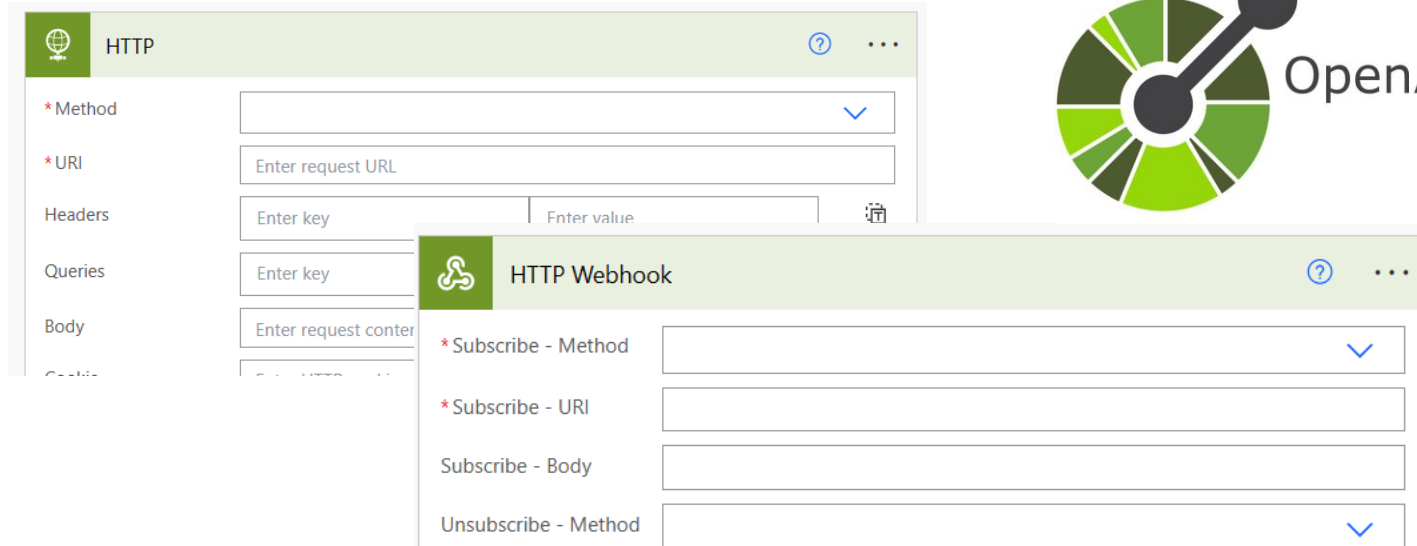
Would you like more details?



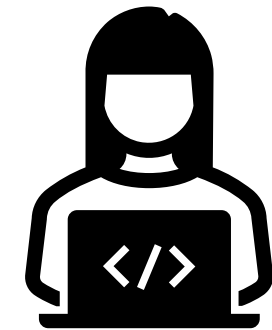
[Using Logic Apps to orchestrate a complex video processing \(...\)](#)

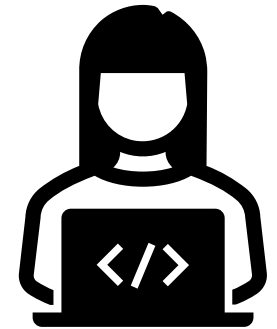
THE GLUE MAKES IT STICK

APIS  CONNECTORS  WEBHOOKS  MESSAGE QUEUES  EVENT MGMT



The screenshot displays two configuration panels from an API management console. The top panel is titled 'HTTP' and includes fields for 'Method', 'URI' (with a placeholder 'Enter request URL'), 'Headers' (with 'Enter key' and 'Enter value' sub-fields), 'Queries' (with 'Enter key'), and 'Body' (with 'Enter request content'). The bottom panel is titled 'HTTP Webhook' and includes fields for 'Subscribe - Method', 'Subscribe - URI', 'Subscribe - Body', and 'Unsubscribe - Method'. Both panels have a green header bar with a globe icon and a question mark icon.



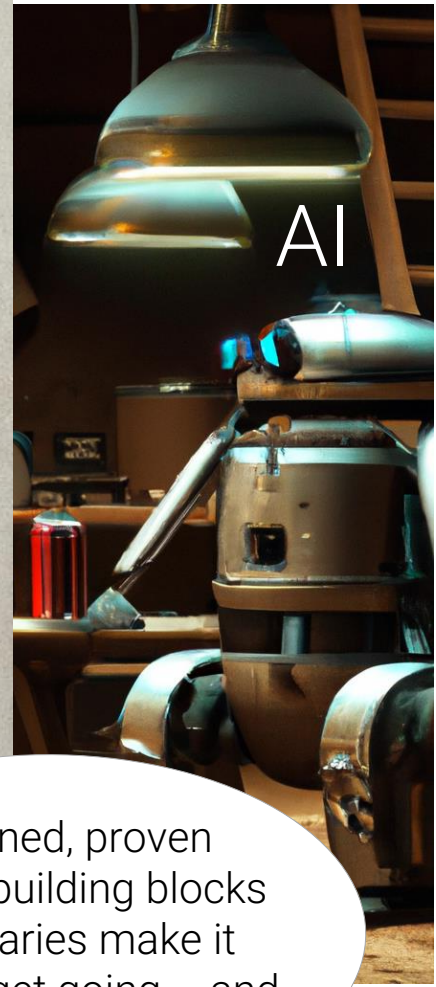




As a BIZ decision maker,
I am still worried all the
time.

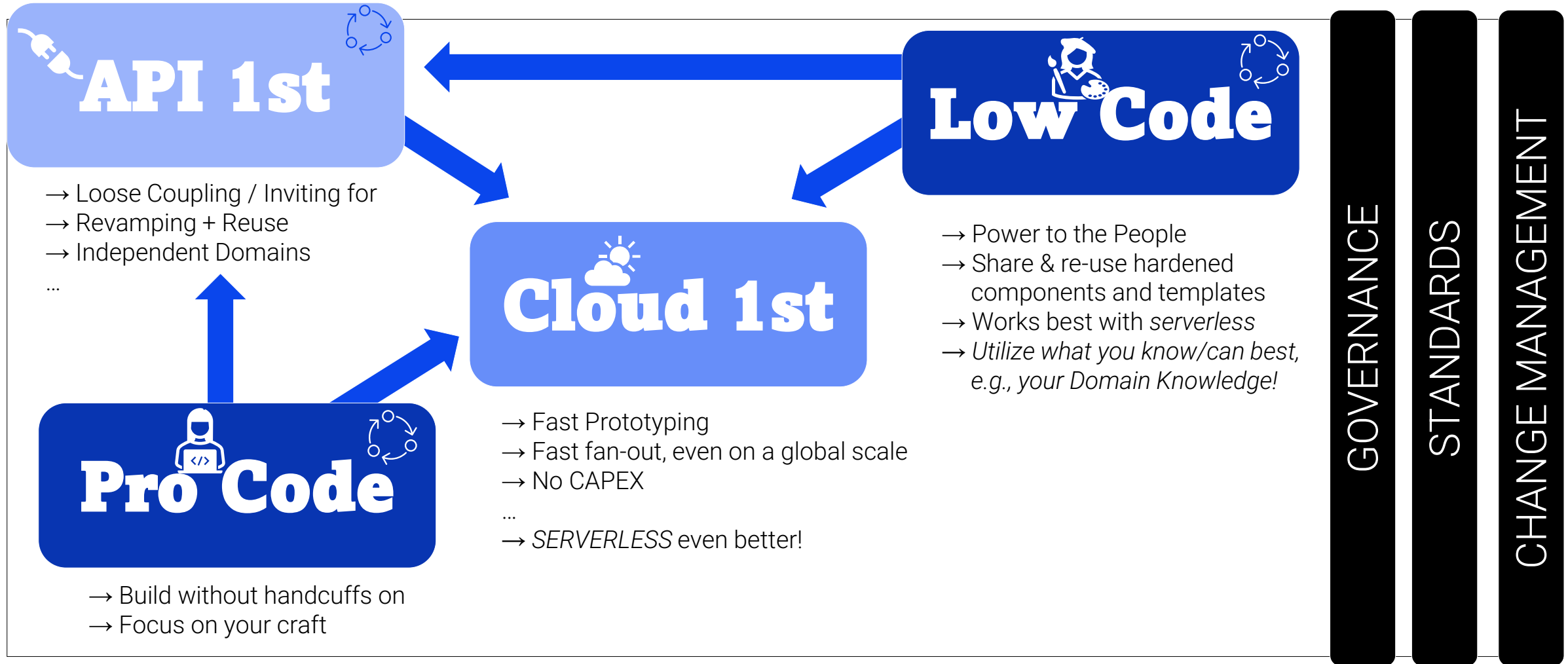
At least the
cross-functional
teams are complaining
a lot less now.

Hardened, proven
solution building blocks
and libraries make it
easier to get going – and
safer, too!



REALITY IS A BIT
MORE COMPLEX THAN THAT

The "Secret Sauce" for success of Fusion Teams



OneDrive

SharePoint

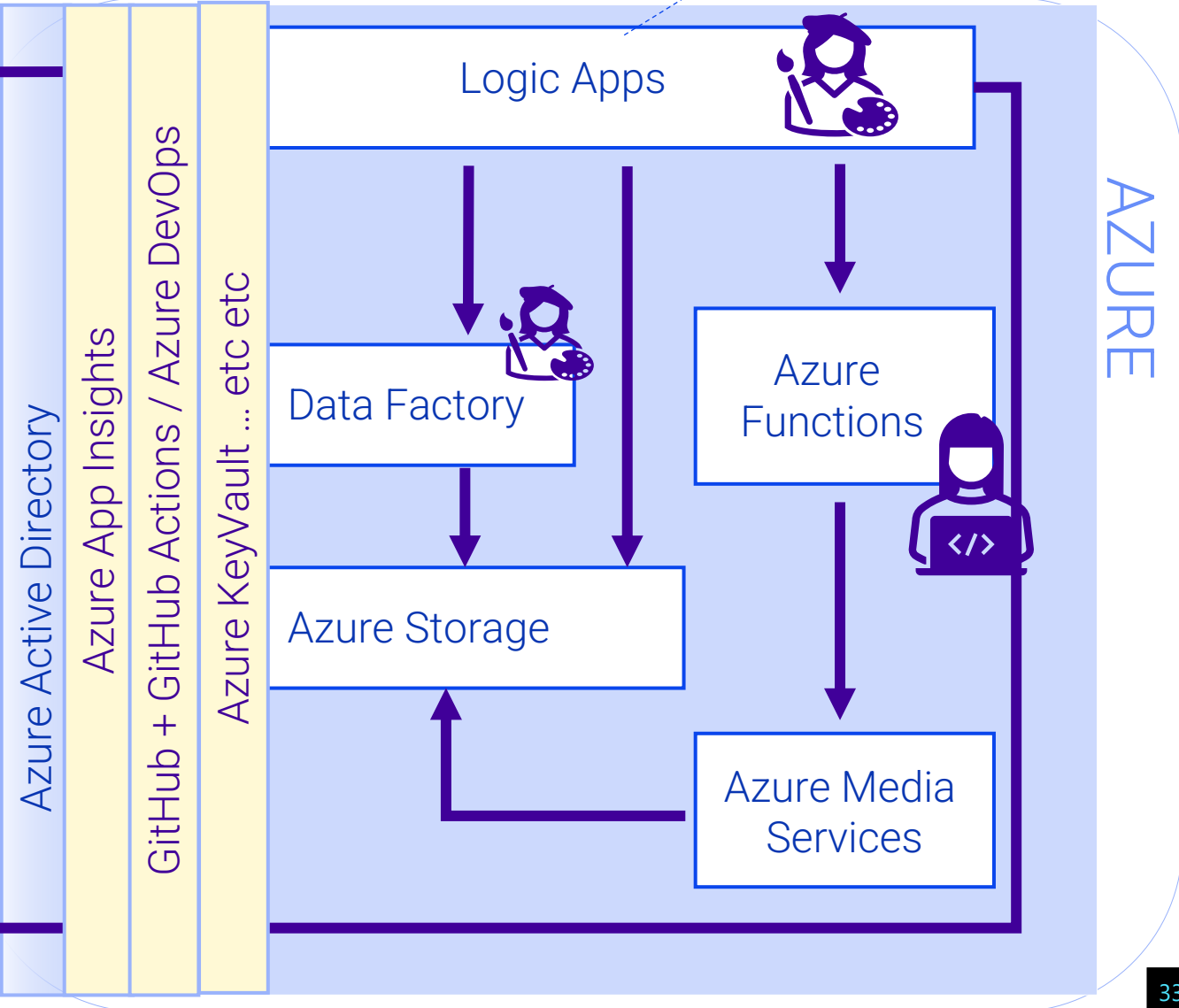
Power Automate

Dropbox



MS Teams

Slack



INTEGRATION PATTERNS ARE IMPORTANT, TOO.

Sidecar, Circuit Breaker and so on.

Be it Low-Code oder Pro-Code, Architecture Decisions always have a grand impact on cost, sustainability, reliability, security and more.

But that is a story for a different time.

Get more relevant links and learning resources here:

platformeconomies.com/azsummit23