



Automating Provisioning For Your Digital Workplace: With Azure Durable Functions & Microsoft Graph

SharePoint Saturday Houston 2019

Vincent Biret

Vincent Biret

Microsoft Office Dev MVP
Azure and Office 365 developer @ 2toLead

@baywet
bit.ly/vince365



Welcome to the 10th Annual HOU365 Saturday

Formerly SharePoint Saturday



The background of the slide features several thin, curved lines in shades of gray, some solid and some dashed, creating a modern, abstract design. A blue speech bubble shape is positioned on the left side, containing the title text.

Attendee Checklist

- Turn phones off or set the ringer to vibrate
- Make FOLLOW @SPSHOU on Twitter
- Make sure you are in the correct room
- SPSHOU Updates #HOU365
- Bingo Cards are on the schedule
- Visit Vendors to complete Bingo Cards
- Enjoy the presentations!



Where is the Social?

- Drinks will be served during the Closing Session and Raffle
- 



Audience?

Agenda

- Introduction
- User Interface choices
- Automation choices
- Azure Durable Functions
- SharePoint specifics
- Microsoft Graph
- Conclusion

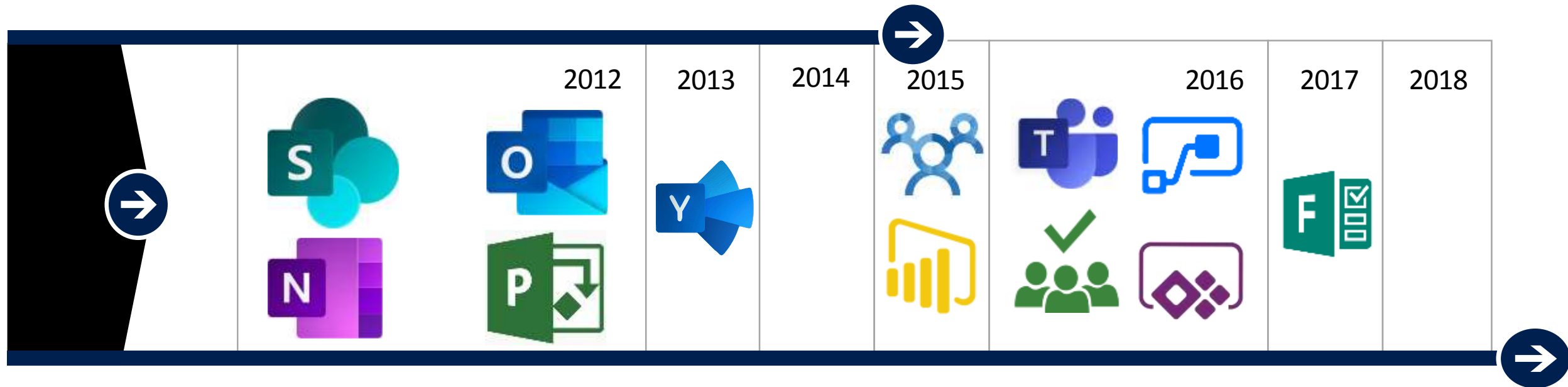
Ready?



Context

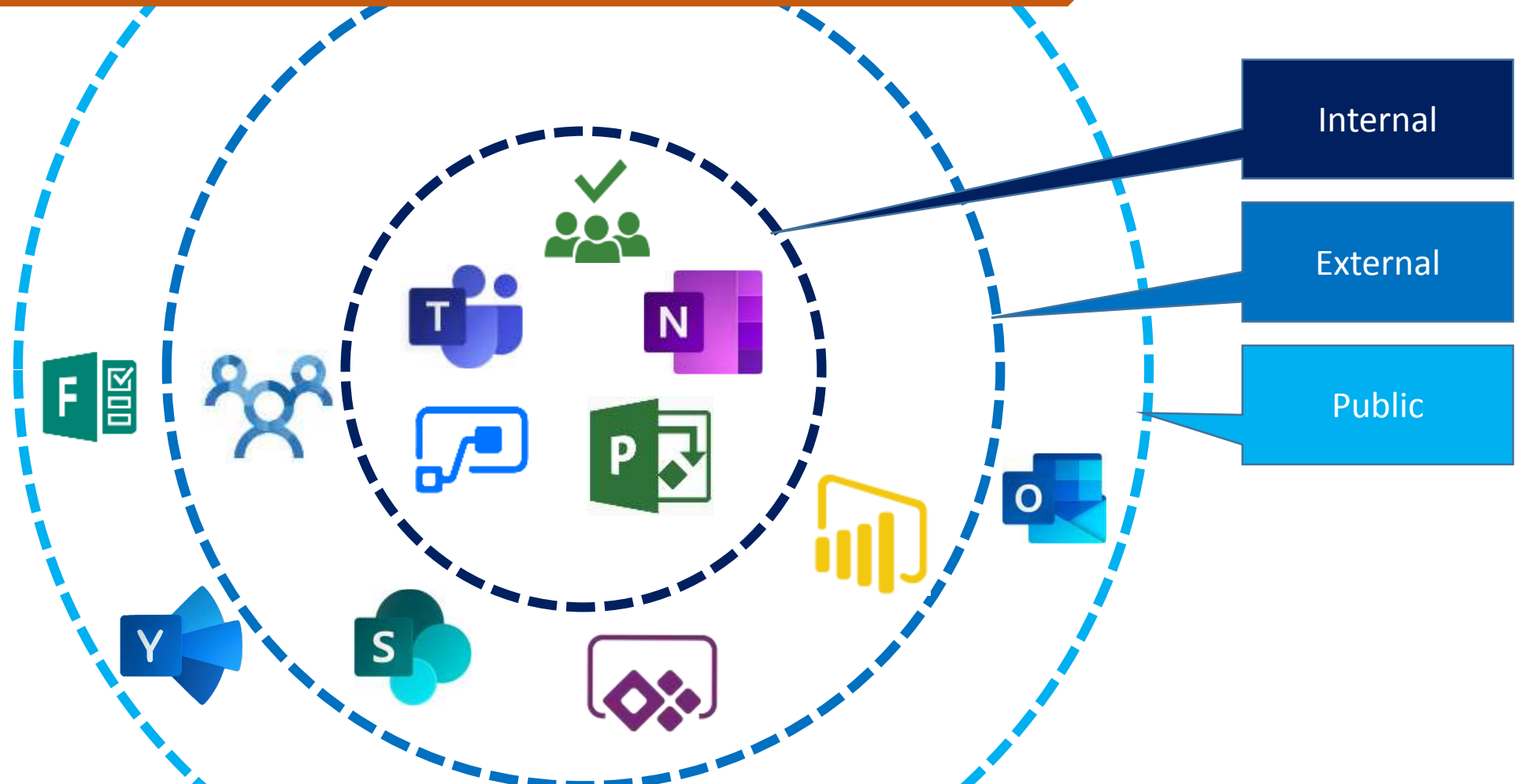


Microsoft 365 Ecosystem



Over the last years Microsoft focused on delivering more specialized tools

Different tools for boundaries



It's important to consider the right tool for the right boundary, and your automation options must account for it as well.

The effects of more specialized tools

Pros

- Improved users efficiency
- Better capabilities
- Faster releases

Cons

- Context Switching ✓
- ~~Licensing complexity~~
- Managing multiple API's ✓
- Administrative lifecycle tasks

The first rule of licensing is we don't talk about licensing

Overall users are happier with better tools, but it added stress on IT and power users.

What do we want to automate?

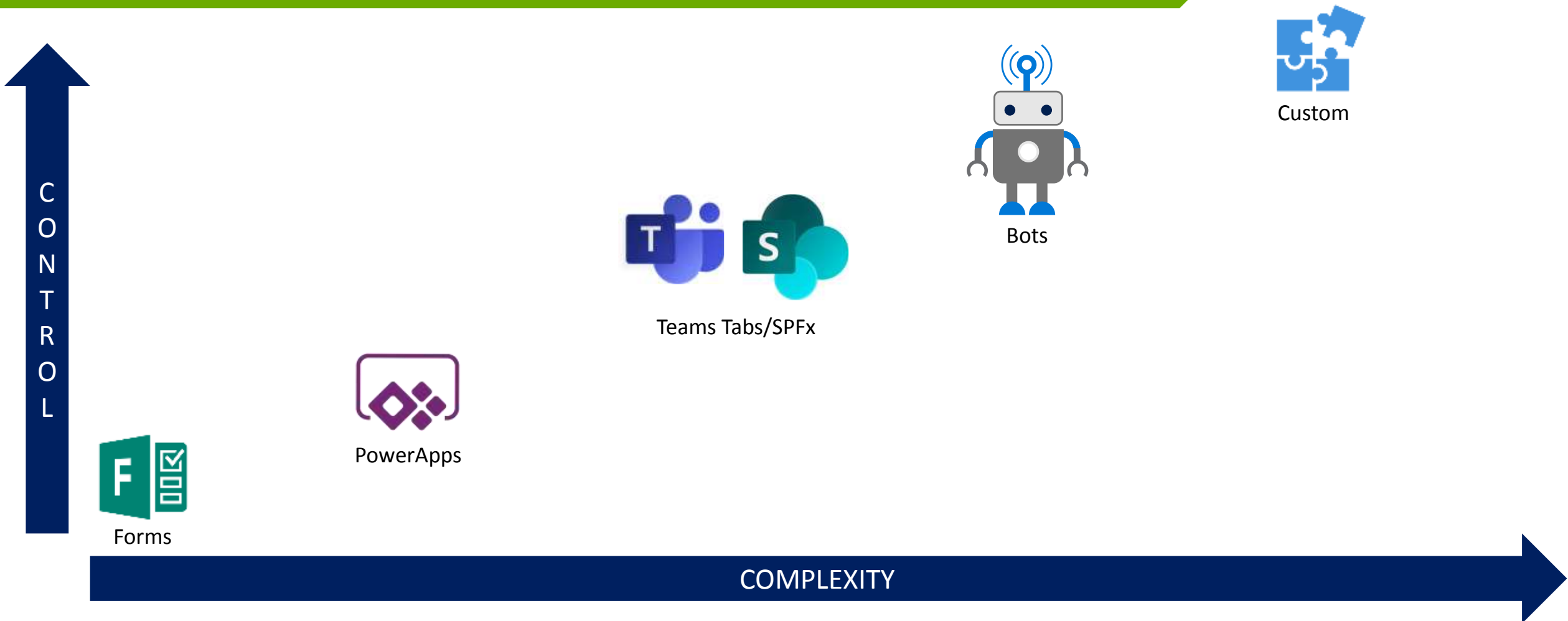
- Create Office 365 group
- Enable Teams on the group
- Update the logo
- Set security
- Create Channels for each project
- Create a templated NoteBook for each project
- Create a Planner Plan for each project
- Add a welcome message
- Add all the tabs
- Create extranet site collection & sites
- Apply a template to it
- ...

Provisioning processes can be quite complex and often require to integrate many technologies together. Thankfully the Microsoft Graph provides most of the APIs.

Building custom User Interfaces in Microsoft 365



Available choices when building custom UI



When building custom user interfaces for Office 365, the trick is to find the right balance.

Benefits of going SPFx/Teams

- Modern technologies (react, typescript...)
- Open source libraries (PnP & more)
- Integrated AuthN & AuthZ
- Integrated hosting
- Discoverable
- High flexibility/control

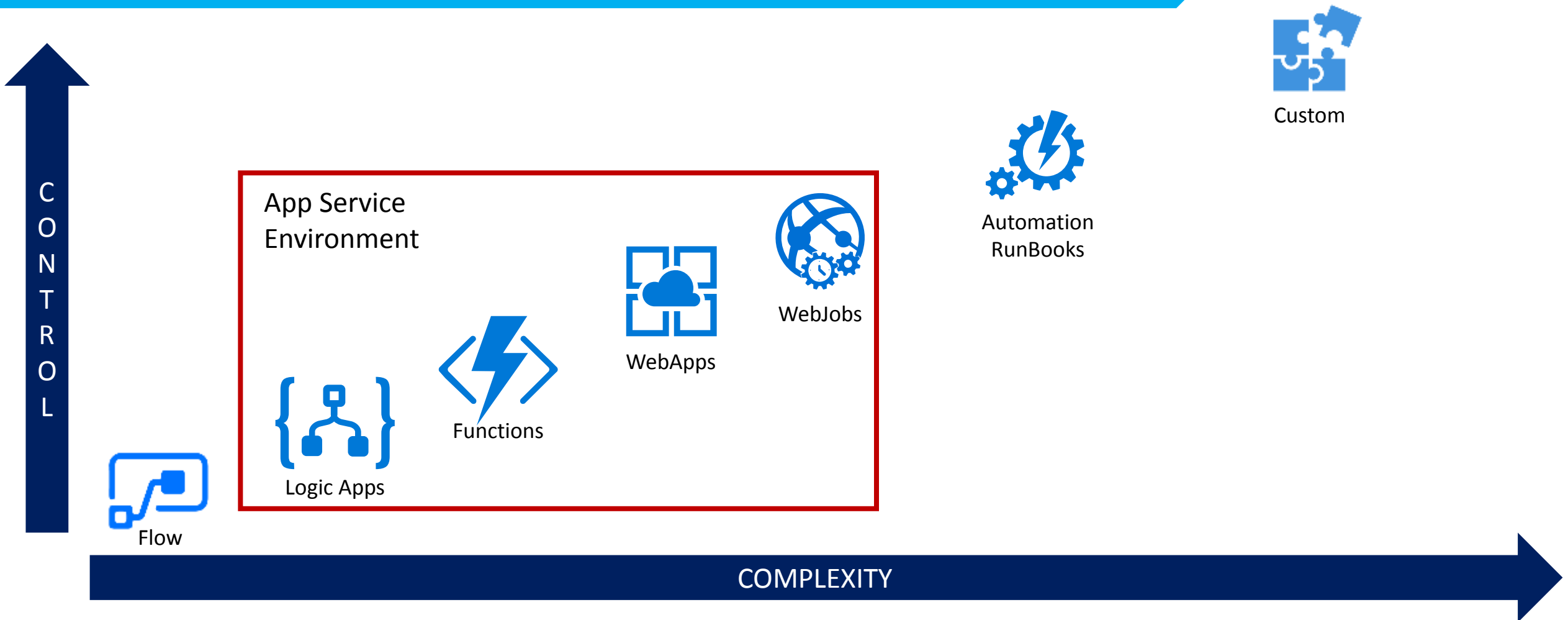
Demo

UI of the provisioning solution



Automating in Office 365

Available choices when automating O365



Azure Functions is a good complement to Flow/Logic App. Logic App good transition from Microsoft Flow to move from user centric to enterprise grade

Benefits of App Service Environment

- Built-in auth (Easy-Auth)
- Built-in security (SSL...)
- Autoscale
- API driven (CORS...)
- Built-in connectivity (vnet, vpns...)
- Proven technology (6+ years)
- Easy to get started
- Cheap

ASE is a great tool to automate things for Office 365, especially if you're a dev, with almost no limitations

Demo



Azure Durable Functions

Value proposition of Azure Functions

Pros

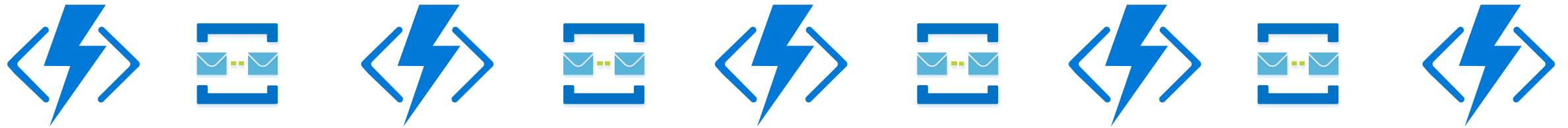
- Cheap
- Fast
- Simple
- Scale extensively

Cons

- Stateless
- Coordination nightmare!
- 10/15 min limit run time

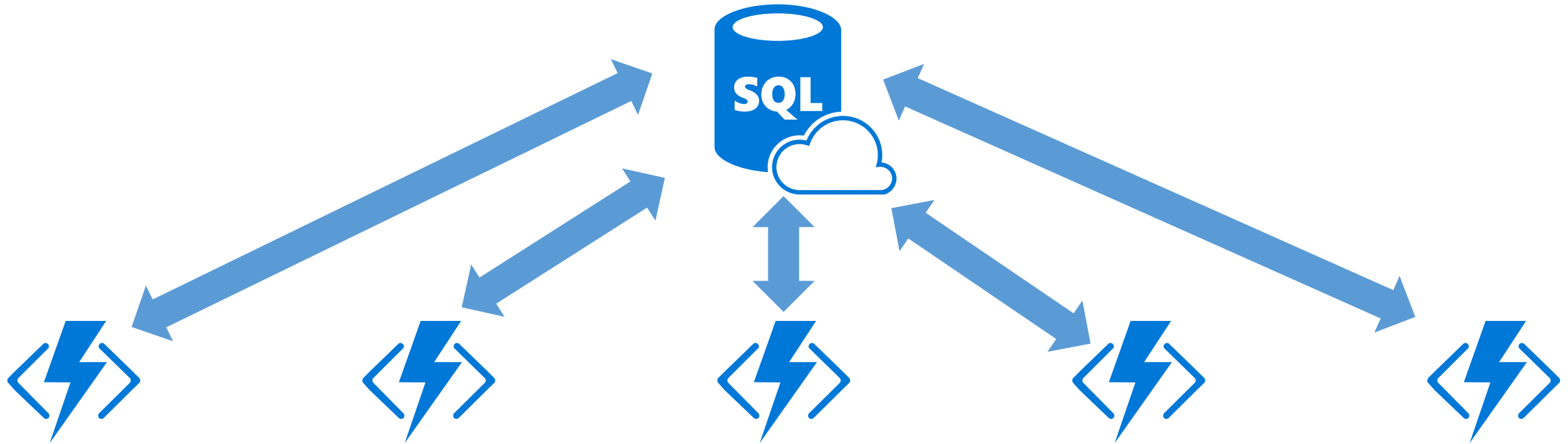
Implementing complex workflows was complex with Azure Functions which are designed for quick and simple activities

Coordinating Azure Functions: Chaining



A common pattern to chain Azure Function was to use queues

Coordinating Azure Functions: State Repo



A common pattern to chain Azure Function was to use a state repository, each function triggering the next one

TRYING TO COORDINATE MANY FUNCTIONS



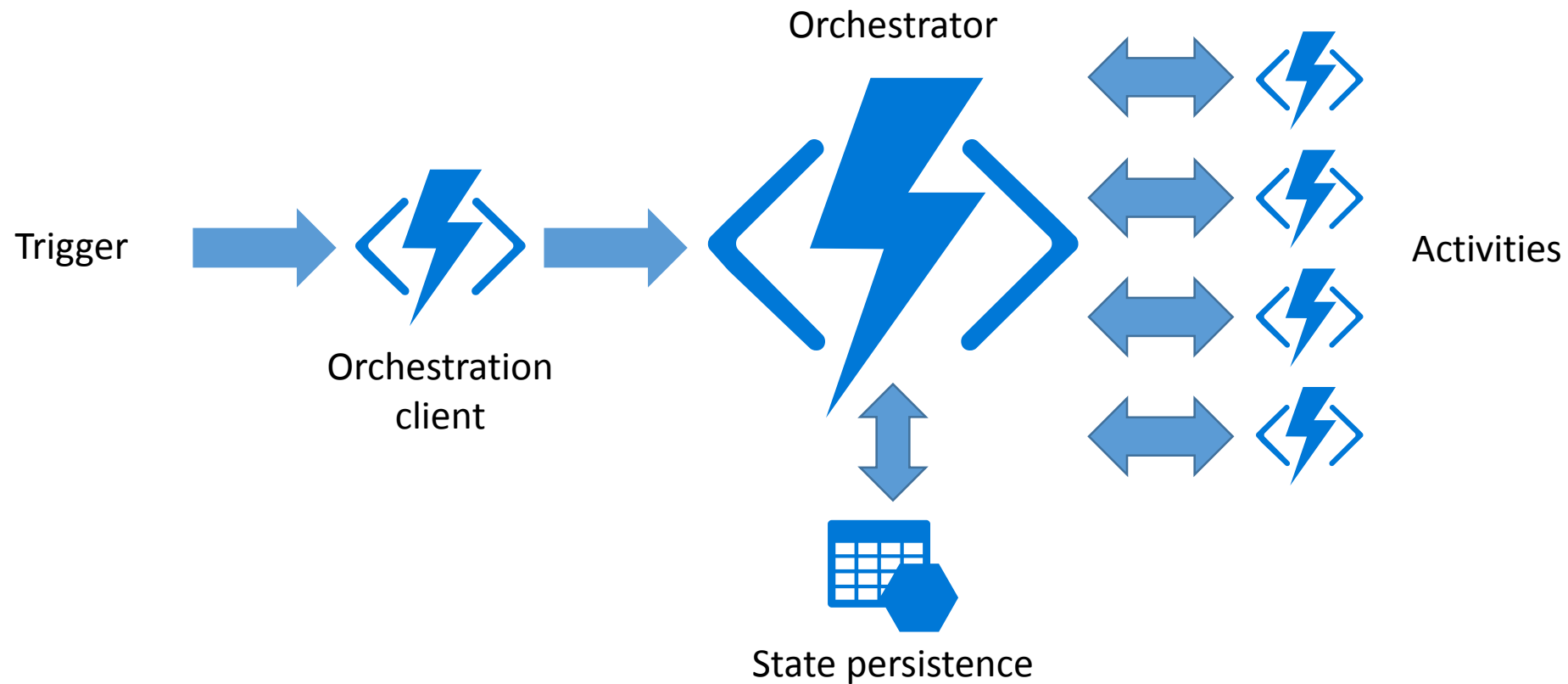
The pain of coordinating Azure Functions grows exponentially with the number of functions.

Introducing Durable Functions

- All the benefits of « regular » Functions
- Code defined workflow
- Automates state management
- Reports status
- Enables new design patterns
 - Fan out/Fan-in
 - Chaining
 - Monitoring

Durable Functions remove a lot of pain introduced with regular functions

High level overview



The Orchestrator leverages a decorator design pattern to « work it's magic »

Important things about Orchestrator

- Orchestrator WILL run multiple time
- Orchestrator ONLY coordinates activities
 - No random
 - No time
 - No async calls
 - Idem potent logic
 - Use orchestrator methods to call activities
- Activities input/output MUST be serializable

Always remember these key principles to avoid strange behavior

Demo



SharePoint Specifics

Authentication & Authorization

- Use Azure Active Directory, not ACS anymore
 - Most of ACS retired a year ago, which is not a good sign for auth part
- Use certificate (not secret) authentication
 - Otherwise, for some endpoints, you'll get a token, and a 403...

╰(´▽`)/

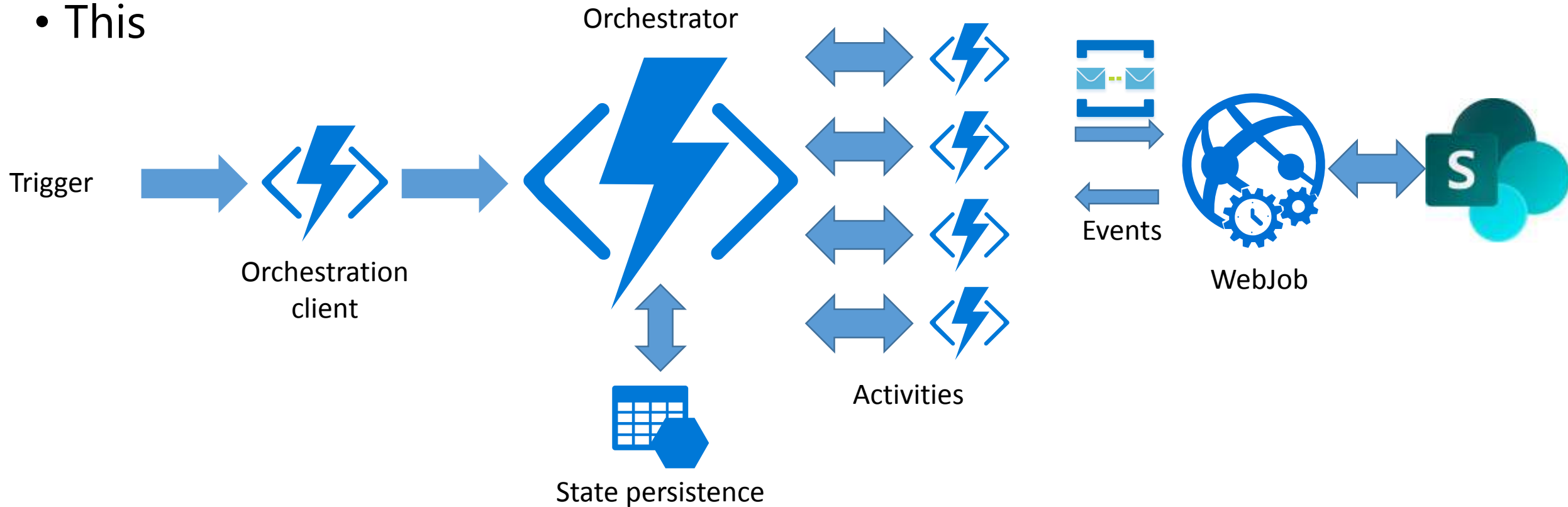
Laggy SDK/CSOM

- CSOM still does not support dot net standard
- Azure Functions are now on dot net core 2.x
- Most libraries moved to dot net standard
- Painful transition ADAL/MSAL on top of that
- Functions/Activities can only run for 10/15 minutes

The situation is currently painful if you want to leverage Azure Functions and work with SharePoint's CSOM

Laggy SDK/CSOM - solutions

- Talk to SharePoint REST APIs & split the code
- This



It is possible to synchronize a webjob and Durable Functions using queues or events. Either to the same orchestrator or a separate one.

Demo

Conclusion



Conclusion

- Lot of time saved
- Consistency of environments
- Self-Service
- Deep integration
- Money saved

Please help me improve this new session



<http://bit.ly/SPSH2019>

It only contains 4 questions, 30 seconds of your time 😊

Thanks!/Questions?



Vincent Biret
Office 365 and Azure
Developer, 2toLead
@baywet
Bit.ly/vince365

Bit.ly/vince365 @baywet slideshare.net/VincentBIRET

