



Håndtér forældede it-systemer – og indfri fremtidens ambitioner

Seminar i samarbejde mellem Devoteam, CLOUDEON - part of Devoteam og Schultz



Dagsorden

- 14.00-14.05 Velkomst og praktisk information v/Simon Svarrer, Direktør og Kristian Kossmann, Salgschef, Schultz
- 14.05-14.40 Strategiske tilgange, der kan anlægges, når legacy skal moderniseres. v/Peter Nørregaard, Expert Director, Devoteam
- 14.40-15.15 Proces og læring fra stort fagapplikationsprojekt. v/Morten Maate, Udviklingschef, Schultz
- 15.15-15.20 Benstræk og kaffe
- 15.20-15.55 Best practice i arbejdet med logiske data modeller, der nedbringer risikoen for systemafhængigheder. v/Frank Mogensen, CTO, CLOUDEON - part of Devoteam
- 15.55-16.00 Afrunding og tak for i dag v/Simon Svarrer, Direktør og Kristian Kossmann, Salgschef, Schultz



Ud af legacy IT? Find den rigtige vej frem

Peter Nørregaard

Expert Director

Architecture & Cloud

Innovative technology consulting for business.

Oplægsholder i dag



Peter Nørregaard

Expert Director, Architecture Cloud
Peter.noerregaard@devoteam.com
T : +45 4178 1316





Agenda

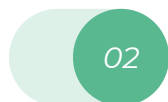


Introduktion til agenda



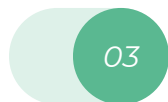
Mål ved at afskaffe legacy

Vi ser på - og udfordrer – målene



Hvornår er et it-system legacy?

Vi se på årsagerne



Veje ud af legacy

De fem veje væk



Spørgsmål



Afrunding og videre til næste indlæg

Typiske mål for at komme ud af legacy

Fra ledelse / direktion / bestyrelse

Vi vil i skyen

Det skal være billigere

Vi vil modernisere

Vi ønsker mere fleksibilitet

Vi vil have standard-systemer

Vi ønsker konsolidering på en ny platform

Vi vil være hurtigere til ny funktionalitet

... som bør udfordres og undersøges nærmere

Typiske mål med at komme ud af legacy ... og til clud

Fra udviklingsafdeling / leverandøren en selv

Spændende
teknologi

On-demand
skalérbarhed

Canary
deployment

Microservices

CI/CD

Gør som
google, netflix,
spotify, ...

... som også bør udfordres og undersøges nærmere



Hvornår er et it-system legacy? Når der opleves et sæt af problemer:



Daglig drift:

- ✓ Dyr i drift
- ✓ Performance flaskehals



Udvikling:

- ✓ Langsomt at udvikle
- ✓ Dyrt at udvikle
- ✓ Svært at teste
- ✓ Langsom release-cyklus
- ✓ Svært at lave digitale produkter i



Samarbejde med leverandør:

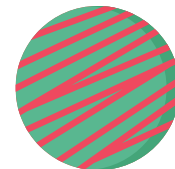
- ✓ Manglende kompetencer hos udviklere
- ✓ Kundens detailstyring ses som nødvendighed
- ✓ Intet sker uden initiativ fra kunden
- ✓ Mistillid til leverandøren

Når strategi til håndtering af legacy skal vælges: Tre vigtige analyser af systemets tilstand

Status på
End-of-life



Graden af
Big ball of
mud



Om systemet
er et Tilted
House

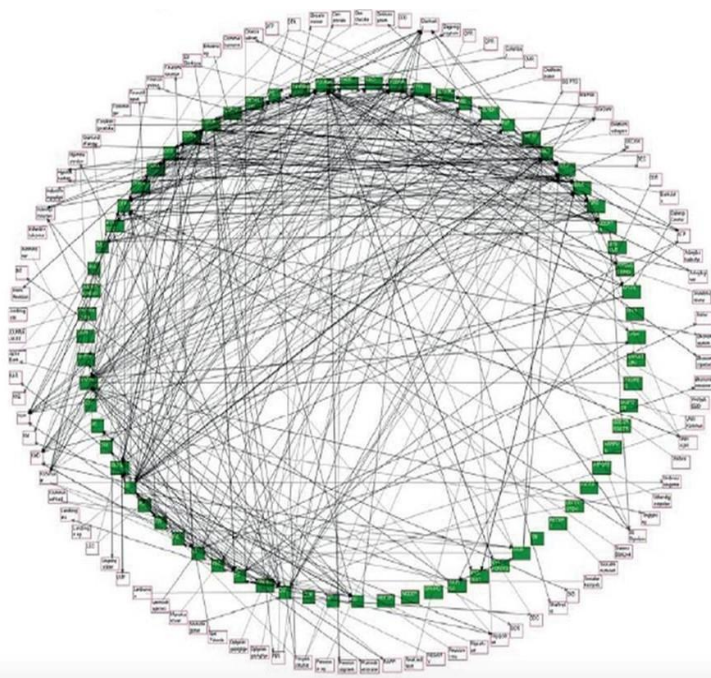
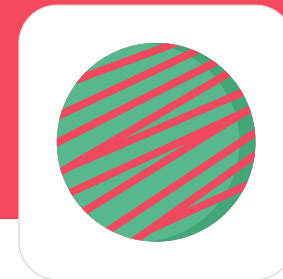


Arkitekturelle anti-patterns

Hvornår er et
it-system
legacy?

Big ball of mud

Når systemet gror til over over tid



**Meet the architectural
anti-pattern *Big ball of
mud or hairball
architecture***

Case: Kompleksitetsanalyse af SKATs it-landskab

Big ball of mud



- **Driver:** Knopskydninger / Feature bloat



- **Driver:** Manglende udviklingskompetencer, -værktøjer og -overblik

Hvornår er et
it-system
legacy?

Tilted house

Når omverdenen forandrer sig over tid og systemet ikke længere er fit-for-purpose



Meet the architectural anti-pattern *Tilted House*

Kilde: Living in a Tilted House: When Continued
Investments in an Information System Will Fail
([Link](#))



Veje ud af legacy

Strategi

0 Genforhandl

- Genforhandl pris
- Genforhandl samarbejde

1 Virtualisér

- Flyt til virtualiserede miljøer. (Lift & Shift)

2 Portér

- Flyt til moderne afviklingsplatform
- Omkod nødvendige dele

3 Modernisér

- Moderniser systemet ved brug af cloud/standardplatforme og -teknologier

4 Standardisér

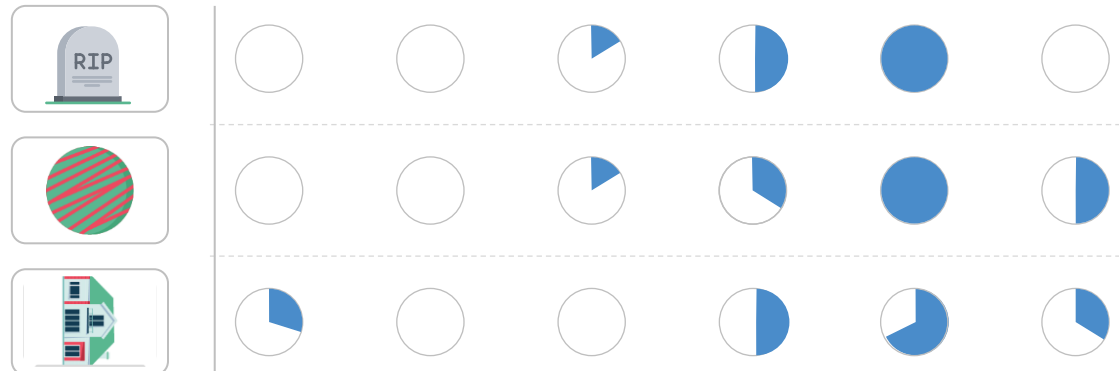
- Lav triage af krav ud i standard-system(er), tilpasning og specialudvikling

5 Reorganisér

- Opdel i *operational backbone* og *digital platform*
- Opdel i legacy og ny platform for fornyelse af legacy

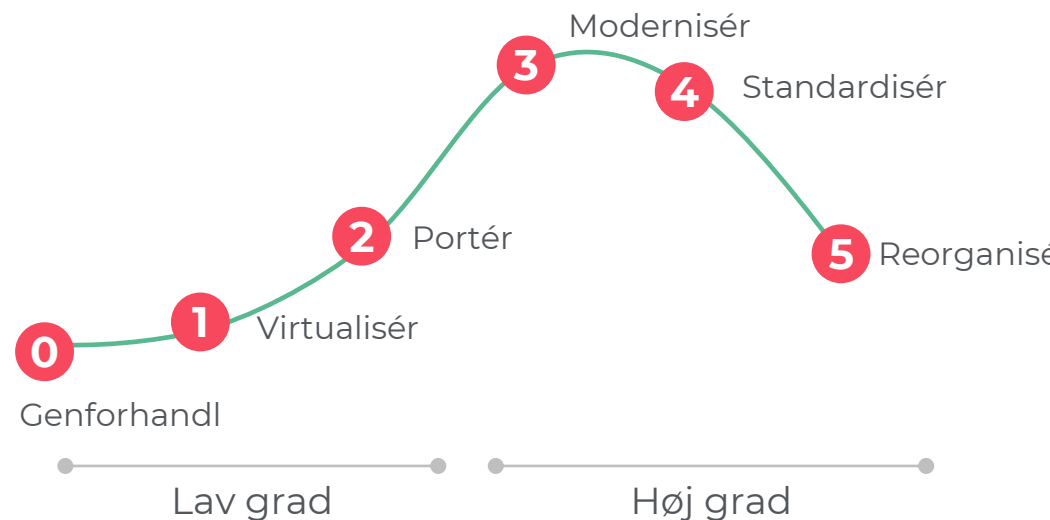
Analyse / strategi matrice

Analysér



Investering (tid og penge)

Høj
Lav



Løses **problemerne** med legacy?

Strategi nr. 5: Reorganisér

- Strategi
- 1. **Opsamlings**
 - Opsamlings om.
 - Gennemgå samfundet
- 2. **Analysér**
 - På et enkelt tidspunkt (J&B-D&B)
- 3. **Planlæg**
 - På et enkelt tidspunkt, udfør nødvendige plan
- 4. **Udfør**
 - Udfør nødvendige plan og brug af cloudleverandere og netværk
- 5. **Overvågn**
 - Overvåg af et standard system, træning i anvendelse
- 6. **Optimer**
 - Optimer leverandere og digital platform
 - Optimer og opgrader for fremtidige behov



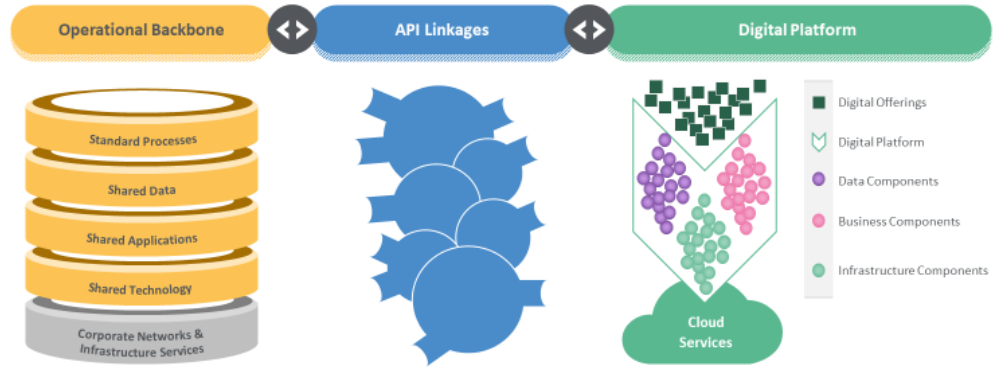
Opdel i *operational backbone* og *digital platform*



Opdel i *legacy* og *ny platform* for fornyelse af *legacy*

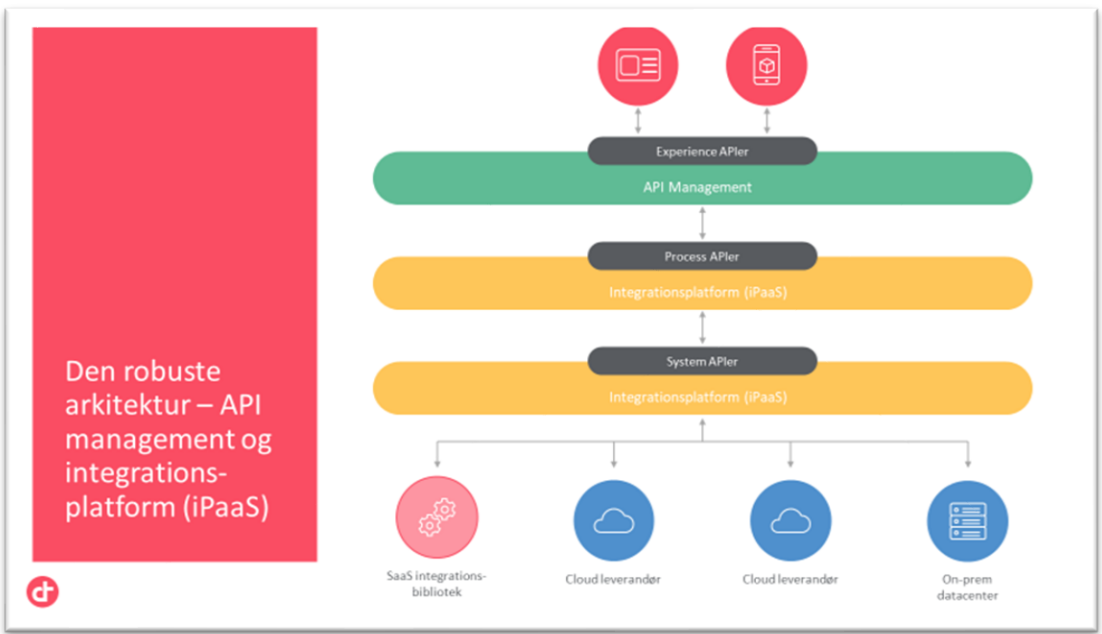


A company that is digitized and digital has two linked technology assets



© Designed for Digital, MIT CSIR, 2019

© Designed for Digital, MIT CSIR, 2019
Jeanne Ross et. al. 2019, [link](#).
(Gentegnet efter originalen på slide 21)



© Devoteam, 2020
Devotalk: Søren Helsted, juni 2020, [link](#)





Opsamling på agenda

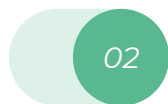


Introduktion til agenda



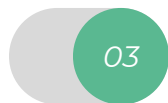
Mål ved at afskaffe legacy

Vi ser på - og udfordrer – målene



Hvornår er et it-system legacy?

Vi se på årsagerne



Veje ud af legacy

De fem veje væk



Spørgsmål



Afrunding og videre til næste indlæg

Konklusioner



Fire gode råd

- ▶ Sæt ind på at forstå de egentlige mål ved at ville ud af legacy
- ▶ Det ér (nogle gange) muligt at tage "legacy" ud af legacy it.
- ▶ Sørg for at nissen ikke flytter med: Vælger I genforhandl / virtualisér / portér bør det kun være første trin på rejsen
- ▶ Udskiftning af legacy er ikke trivielt – sørg for at afsætte tilstrækkelig tid og ressourcer til analyser, refleksion og forankring



Spørgsmål

- ▶ Spørgsmål er velkomne

thank you.

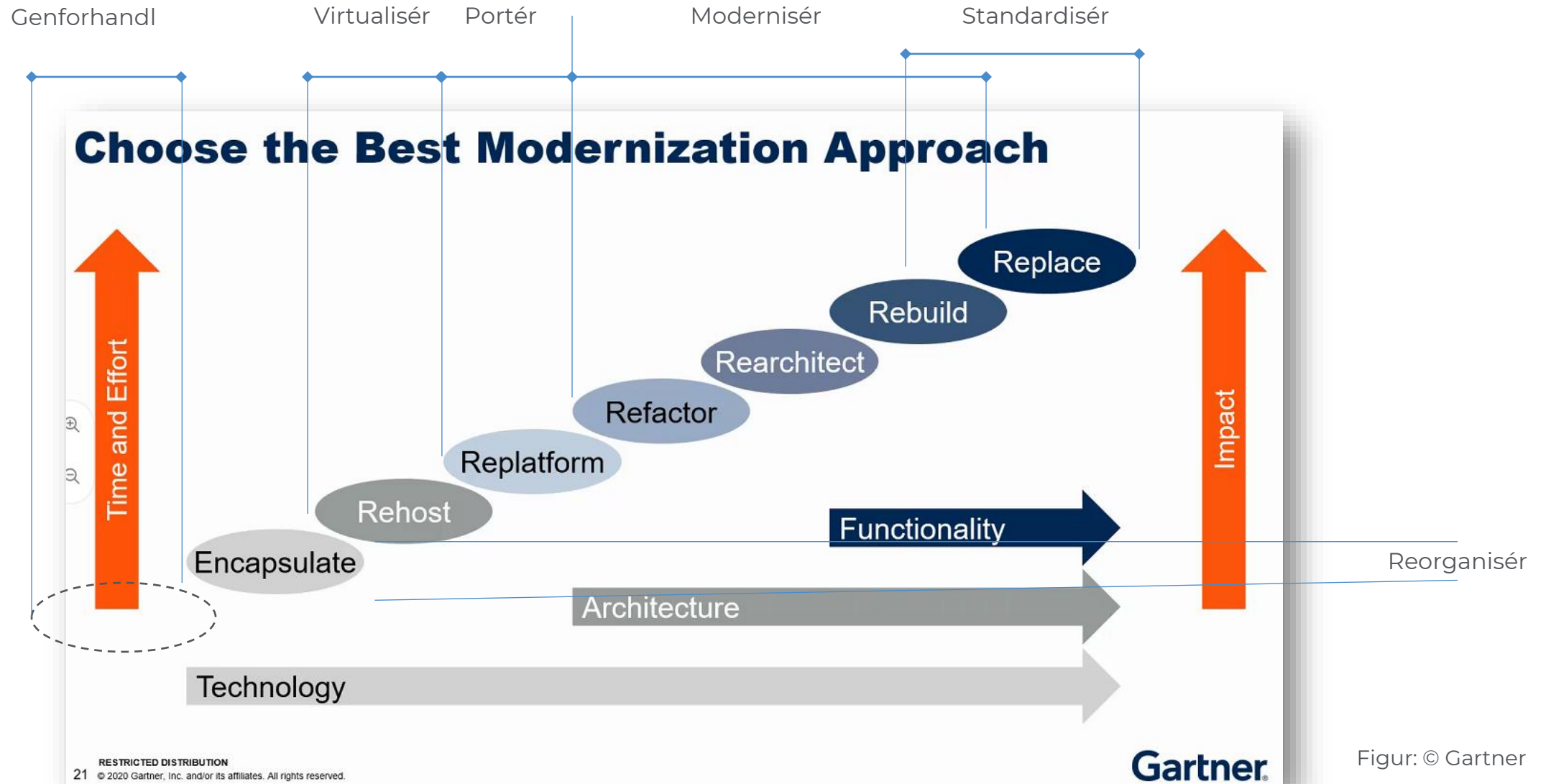
#TechforPeople.



Bonus & back-up slides



Devoteams strategier til håndtering af legacy: Relateret til Gartners 6*R-model



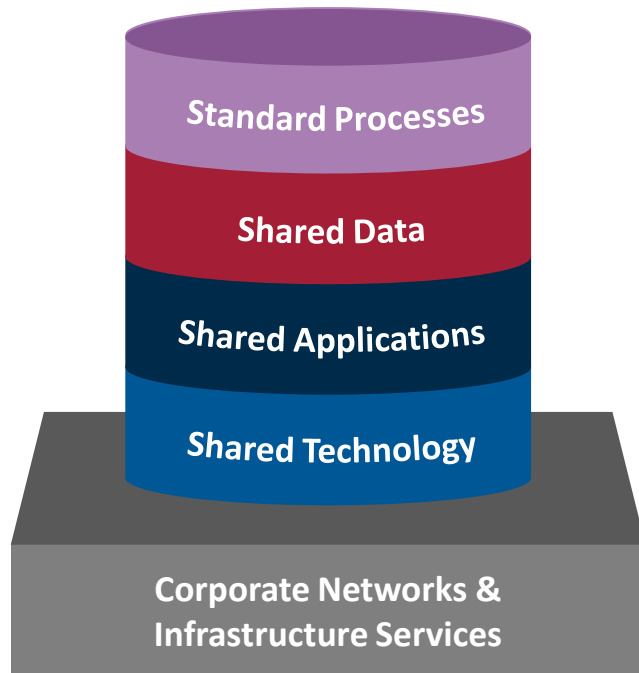
Figur: © Gartner



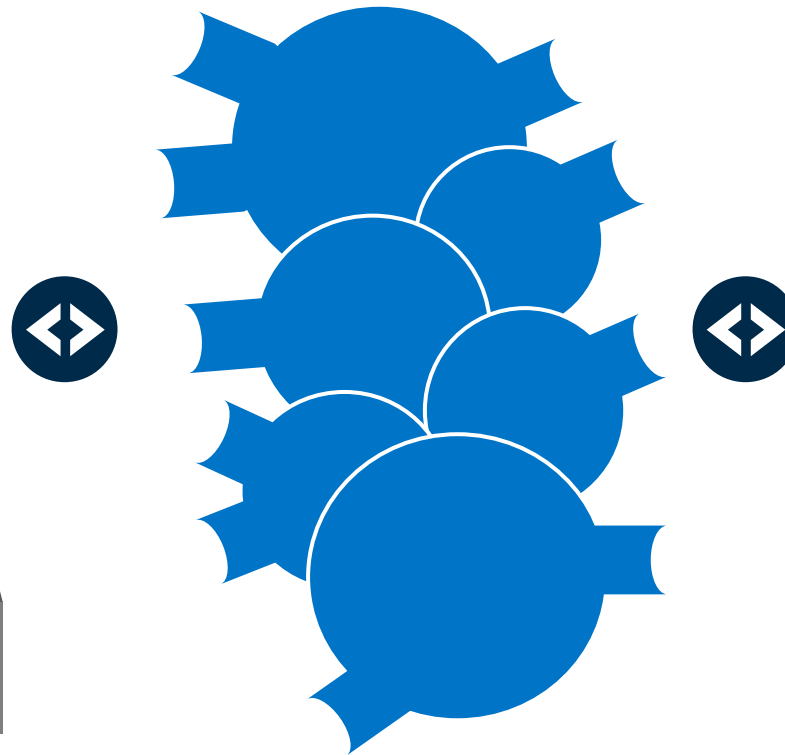
A company that is digitized and digital has two linked technology assets



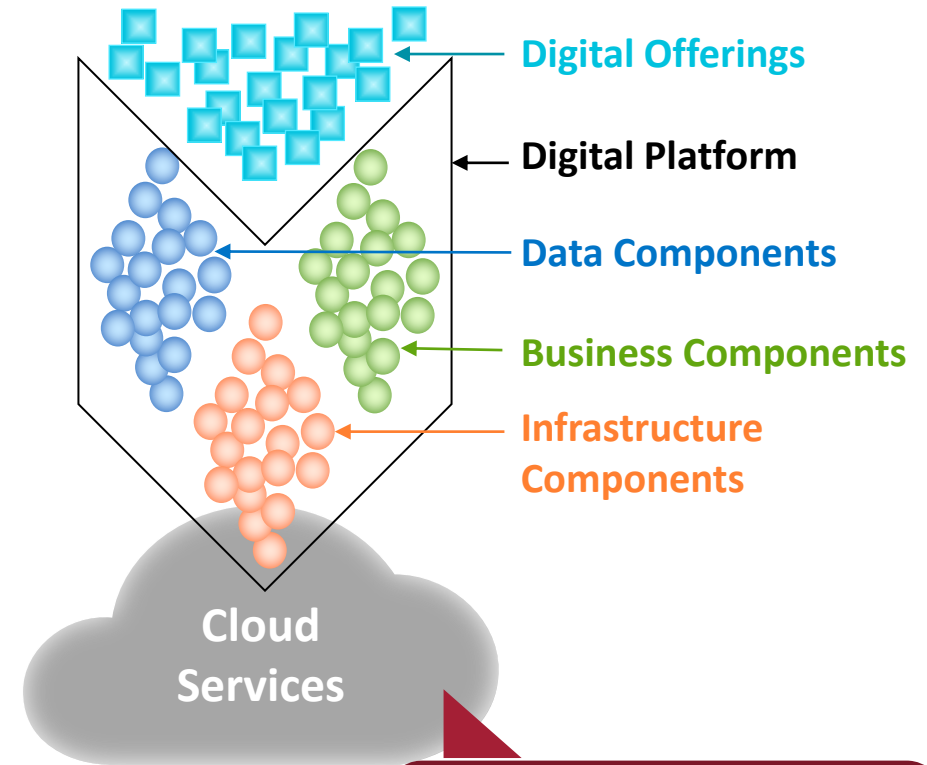
Operational Backbone



API Linkages



Digital Platform



Original slide fra
"Designed for Digital"
© MIT CISR



GØR DET KOMPLEKSE **SIMPELT**



SKAB **KONTROL** IGENNEM
DOKUMENTERET FREMDRIFT

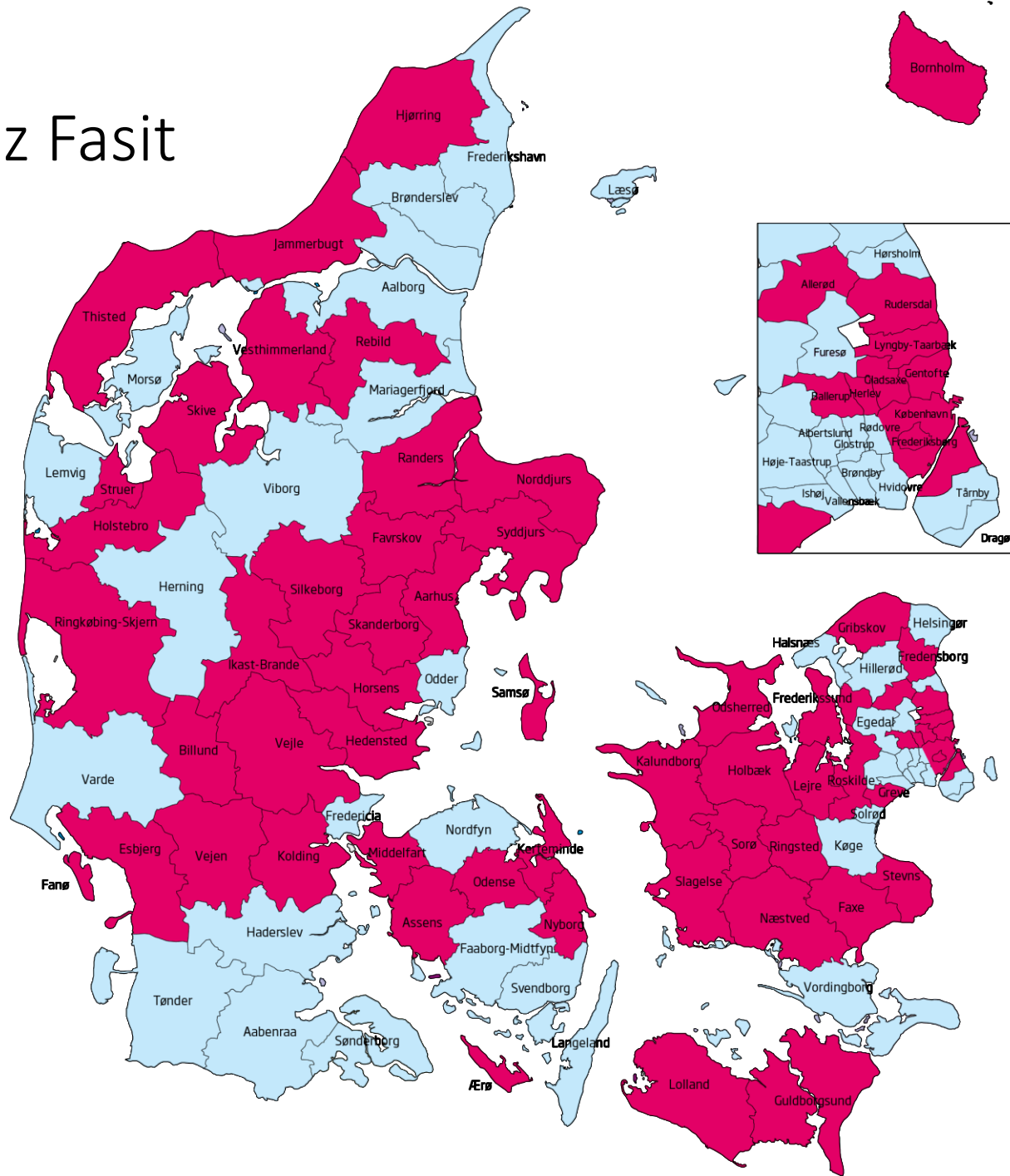


SIKR **BRUGERFEEDBACK** IGENNEM
LØBENDE IDRIFTSÆTTELSER

Agil modernisering: Et case-studie af Schultz Fasit

Morten Maate, It-udviklingschef, Schultz

Schultz Fasit



Kort om casen

Systemet

- Schultz Fasit er med 59 kunder og 14.000 faste daglige brugere markedets foretrukne jobcenterløsning.
- Systemet er forretningskritisk og anvendes intenst i hele forretningsstunden.

Formål med moderniseringen

- Vi har altid moderniseret, men vi gør det nu i et hidtil uset tempo.
- Det primære formål med moderniseringen er at blive frigjort fra den underliggende Microsoft-platform, som i dag er forældet.
- Imens vi moderniserer, vil vi skabe en ny brugeroplevelse.
- Vi moderniserer transitionelt: vi flyver maskinen, mens vi renoverer den.

Mens vi har moderniseret

- har vi idriftsat 17 nye kommuner og ikke tabt en kunde.
- er driftsstabiliteten steget med 50% - ifølge brugerne.
- er tilfredsheden med brugervenligheden steget 8 %.



HEALTH CHECK APEX PROGRAMMET 2021

Der er etableret klare sammenhænge mellem bestyrelsesindstilling (2018), strategi (2020) og programmets nuværende planer, arkitektur og leverancemodel og med tæt involvering af kunderne. Dette er absolut best practice.



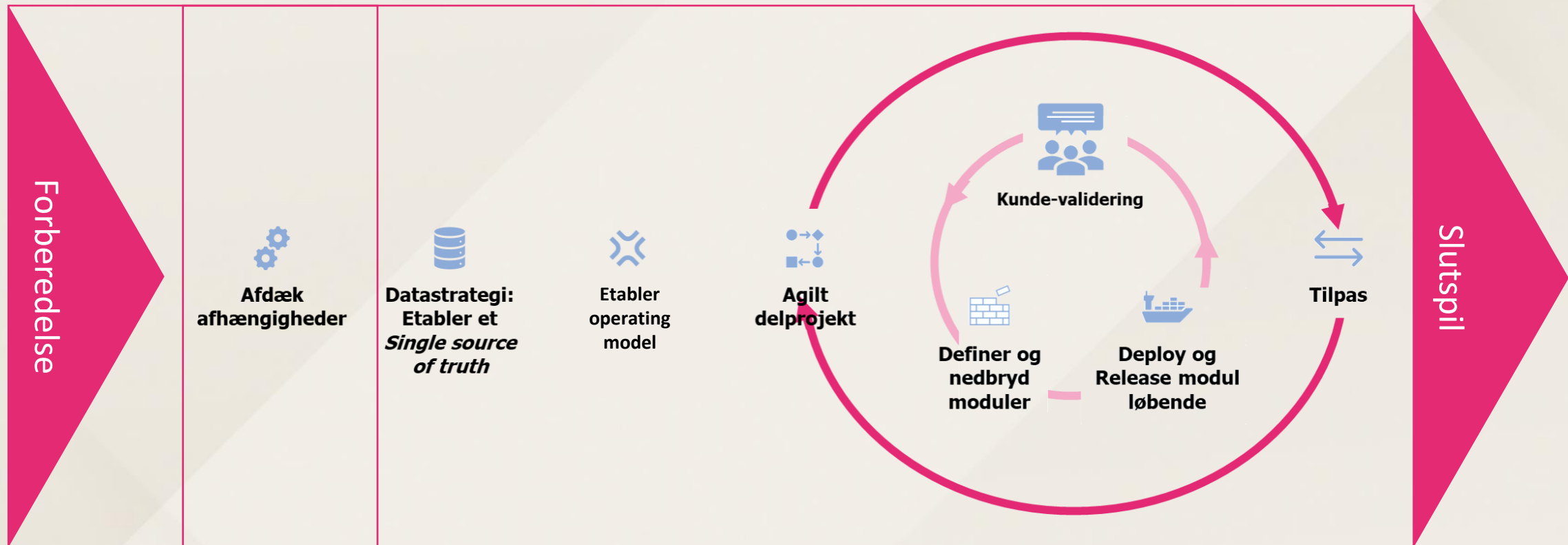
Kenneth Schørring

Partner

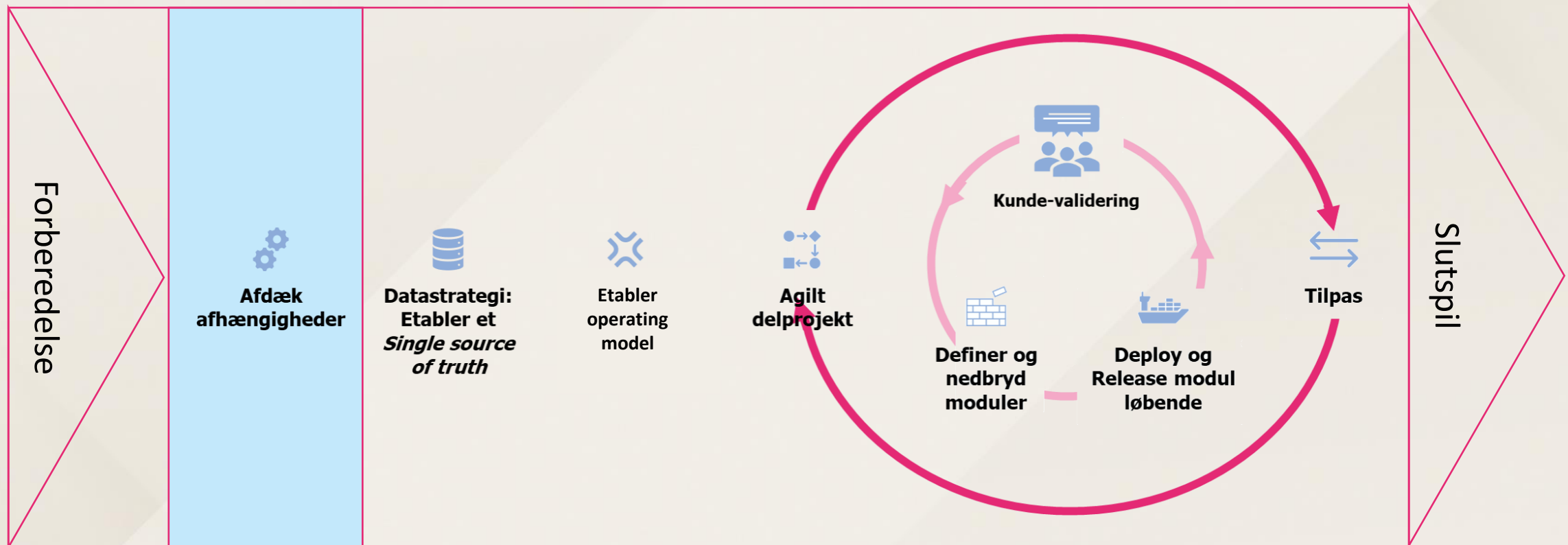
Winspire A/S



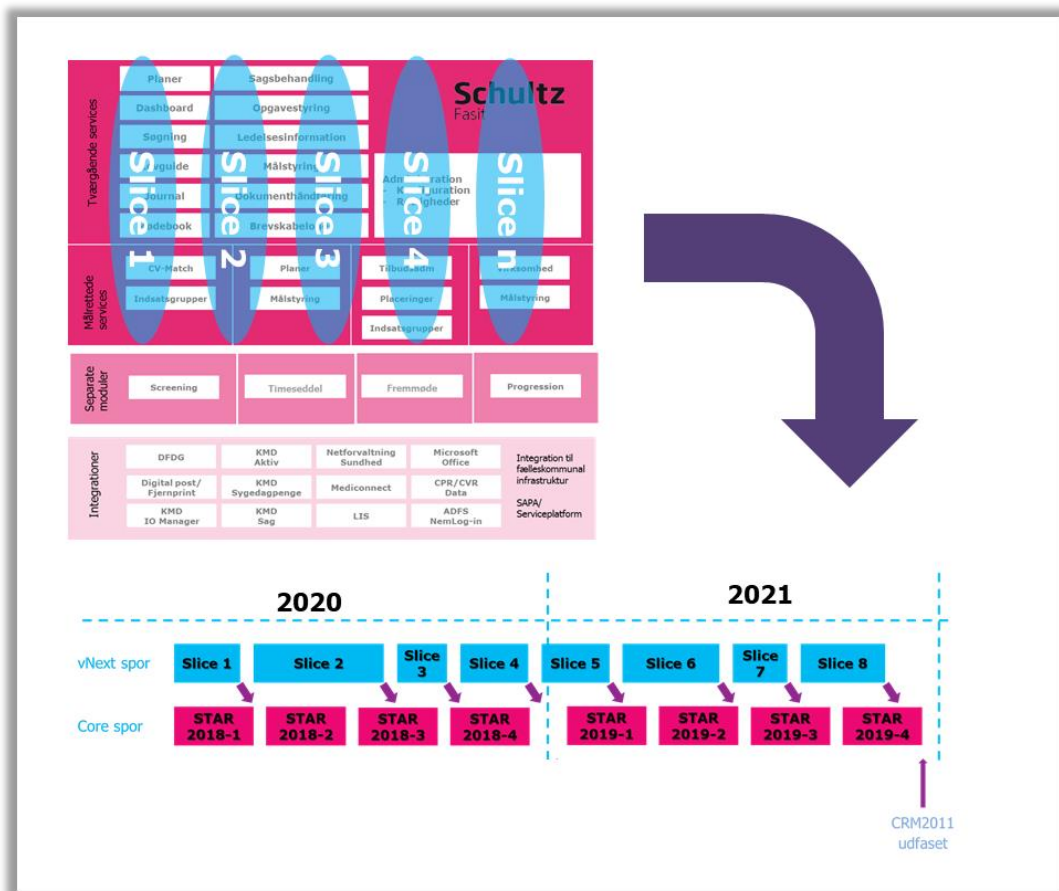
Agil moderniseringsmodel



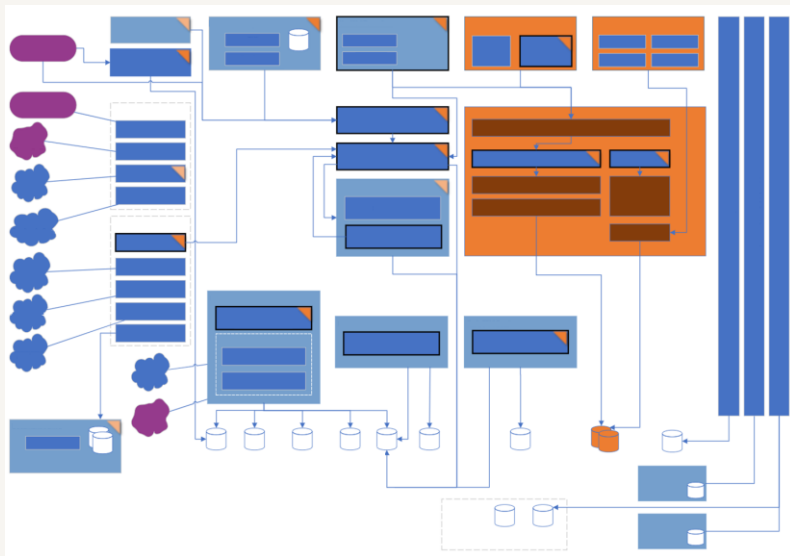
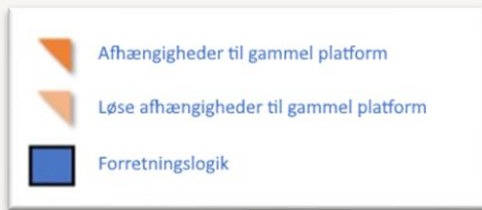
Agil moderniseringsmodel



Afdæk afhængigheder



- Trinvis vertikal migration ved at bruge bounded context
- Nedbrydning af monolit til funktionelle delområder
- Feature maps til at udvinde krav
- Langsigtet mål, mindre og kontinuerlige leverancer
- Planlægning og design af løsning og arkitektur



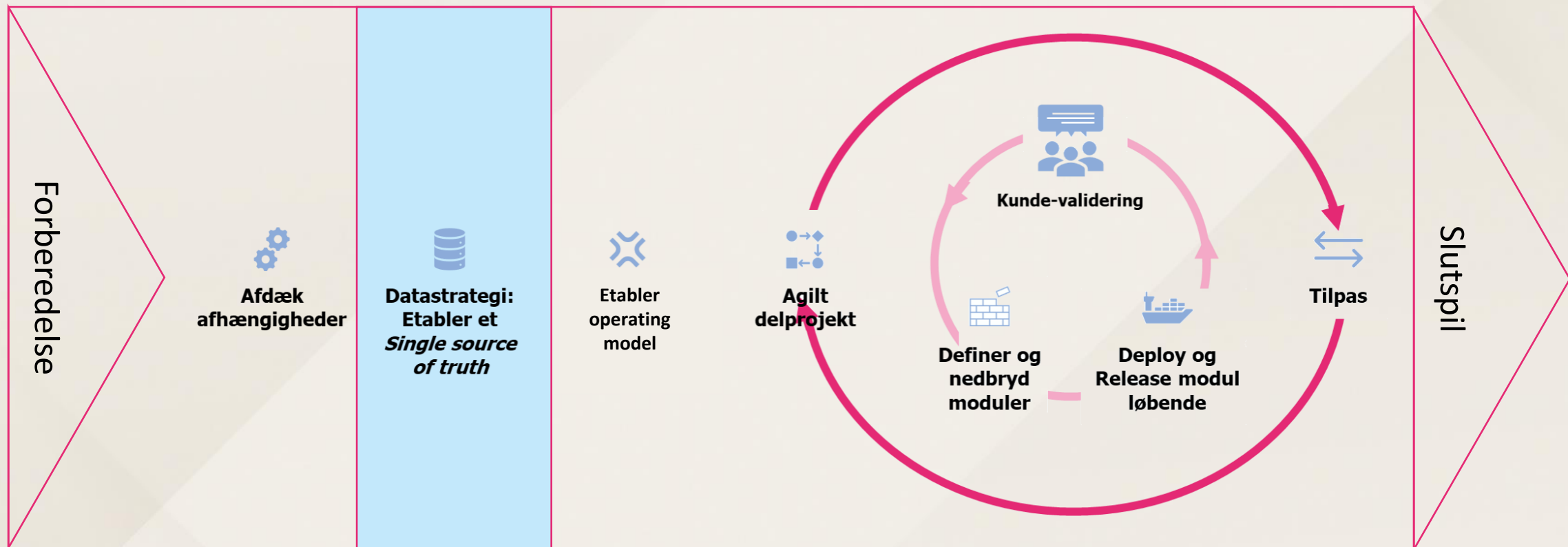
As-is billede af Fasit før moderniseringen, 2018

► Konklusioner i foranalysen (2017)

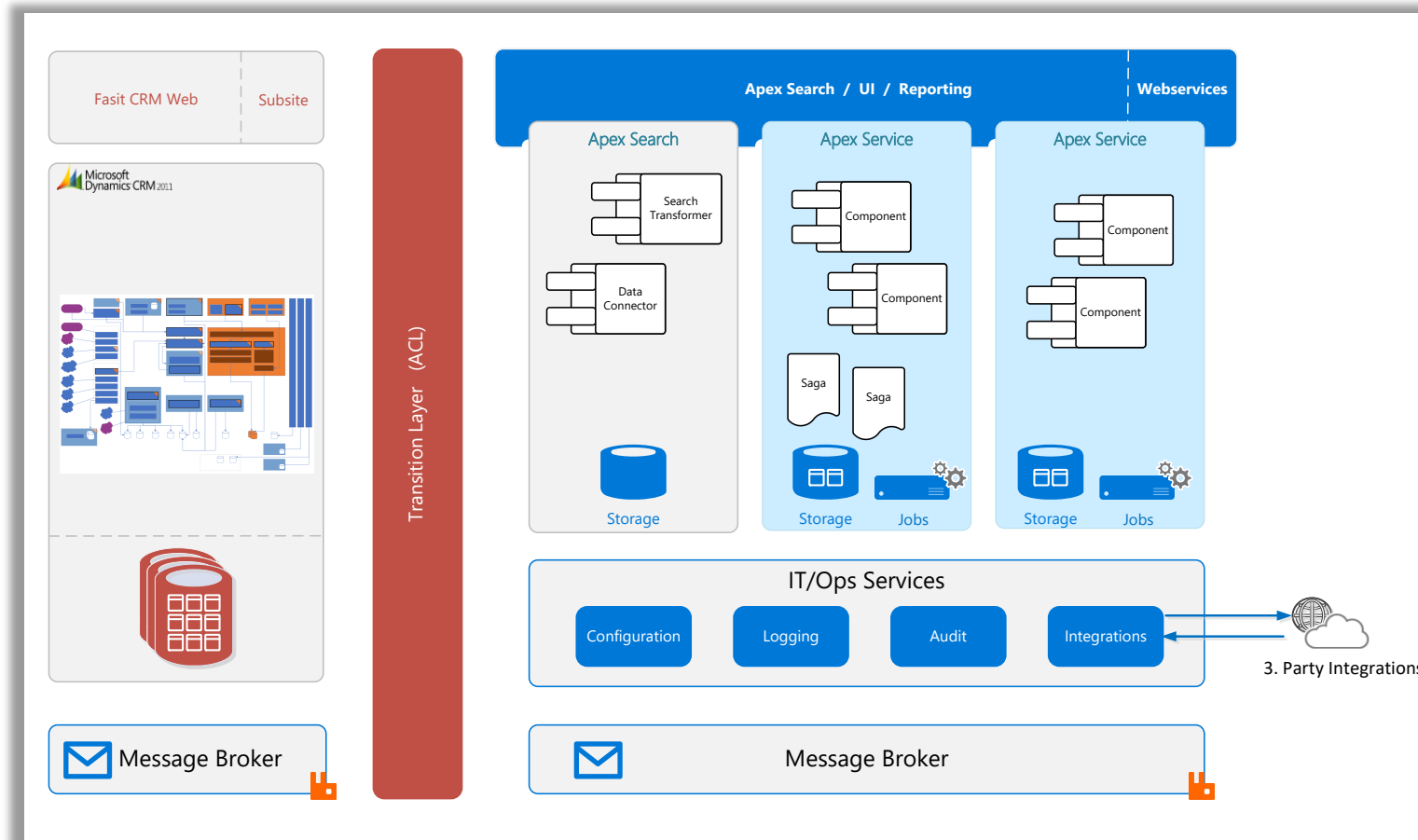
- Udfordringer:
 - Der er afhængigheder mange steder
 - Forretningslogikken er spredt
- Men også gode nyheder...
 - Den gamle platform *kan* udskrives *transitionelt*: Fasit kan leve på ryggen af to platforme.
 - En række komponenter kan videreføres uden modernisering
 - En række teknologier kan videreføres uden modernisering



Agil moderniseringsmodel

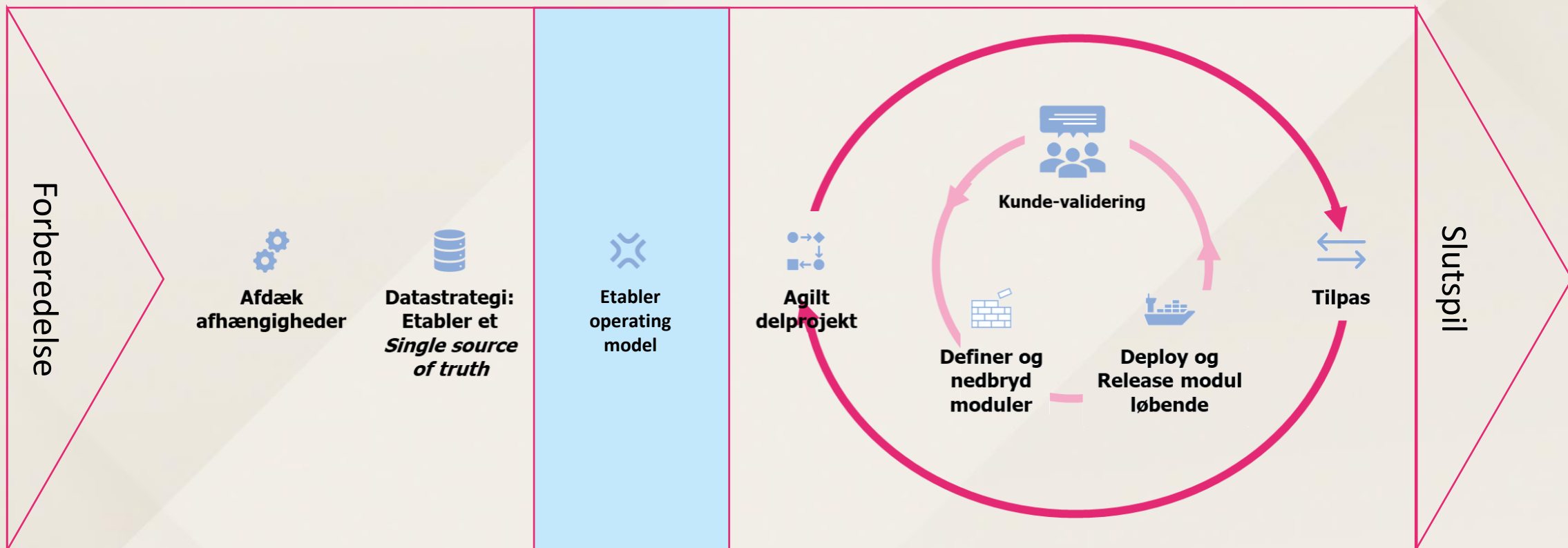


Scope, estimat og projektplan er etableret med udgangspunkt i den fastlagte målarkitektur

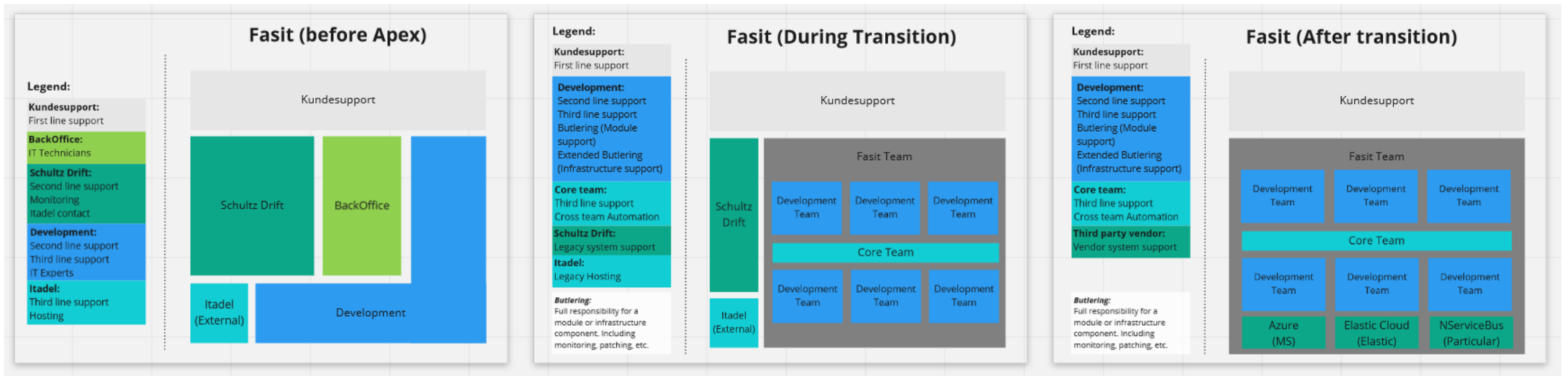


- Målarkitekturen er baseret på moderne arkitekturprincipper.
- Den er baseret på Cloud og anvendelse af standardkomponenter.
- Arkitekturen giver mulighed for, at vi kan skifte driftsleverandør/Cloud.
- Data sink: Første komponent etablerer et *single source of truth* for vores data.
- Der etableres et transitionslag, som gør det muligt at foretage løbende udfasning af den gamle platform.
- Arkitekturen understøtter vores strategi om at release løbende til markedet – hvilket giver lav risiko ift. bl.a. driftsstabilitet. Det understøtter samtidig en bevægelse hos vores kunder ift. at hyppigere lovopdateringer.

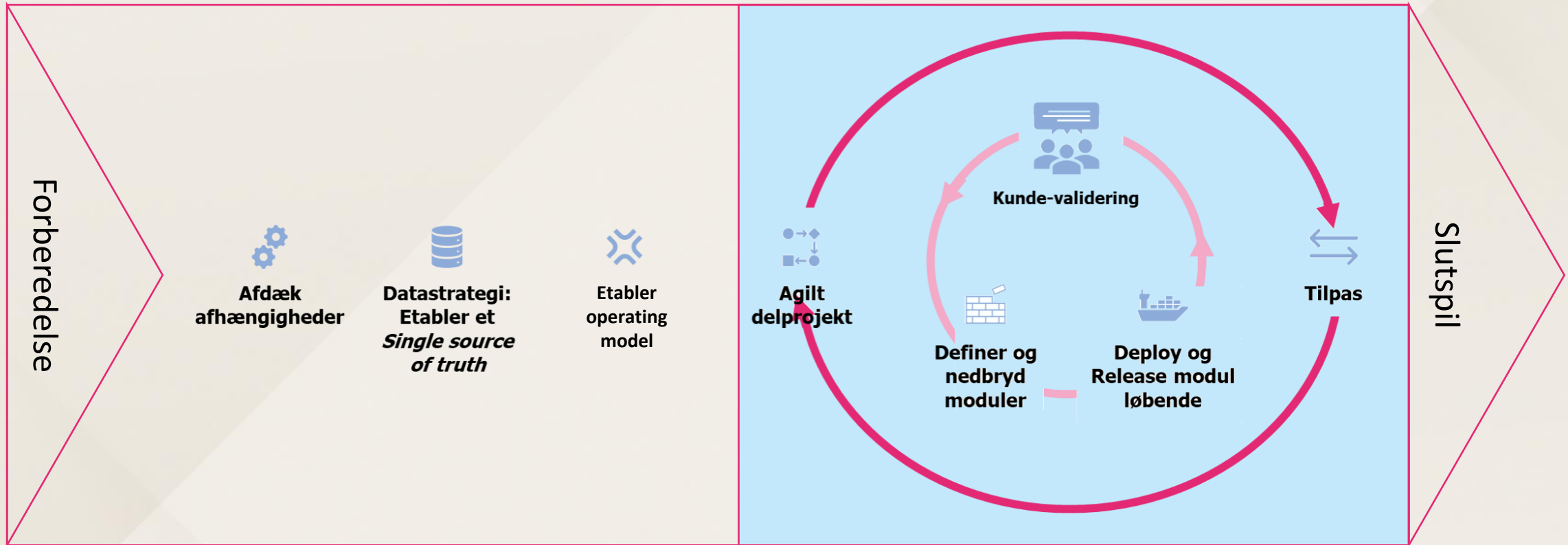
Agil moderniseringsmodel



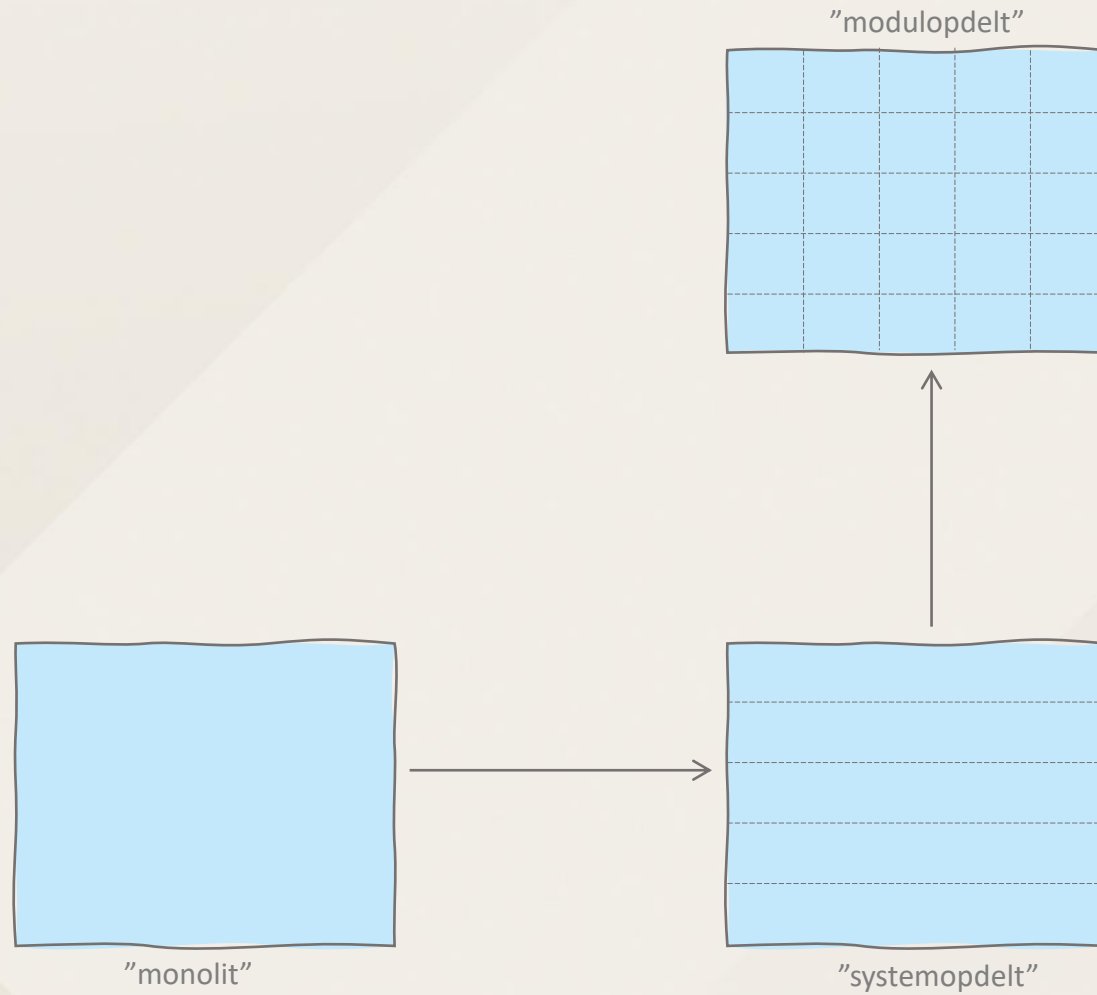
Target Operating Model



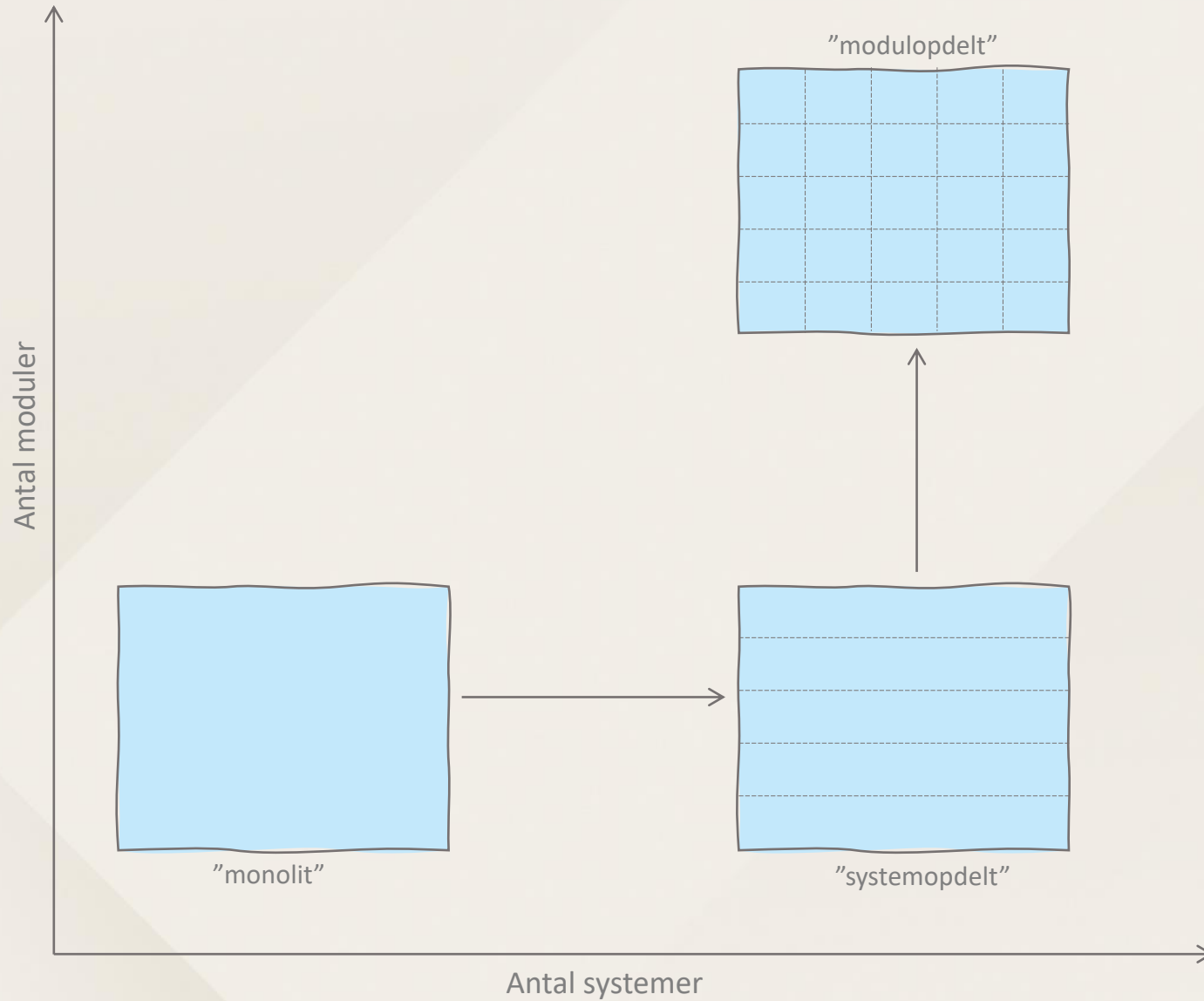
Agil moderniseringsmodel



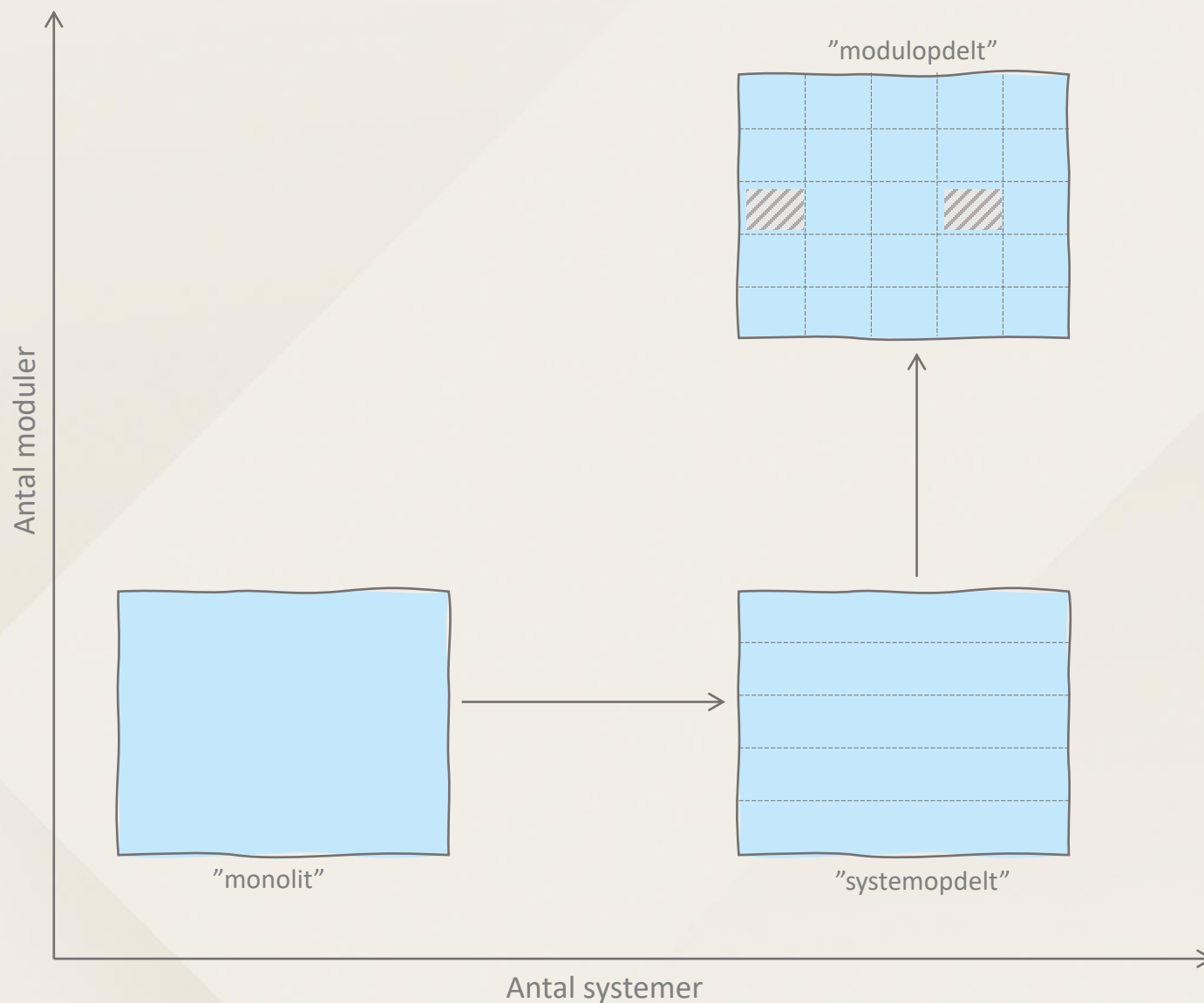
Agil modernisering



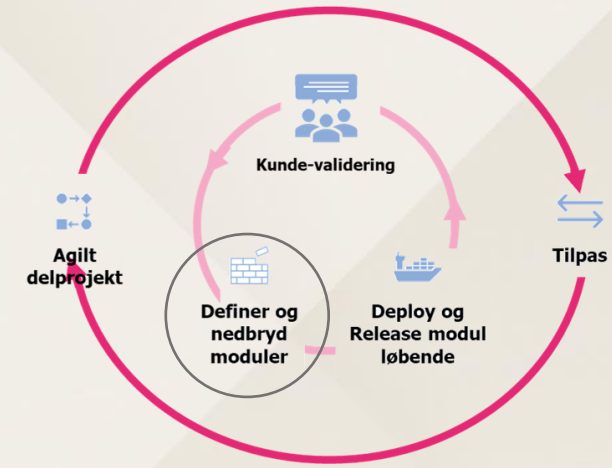
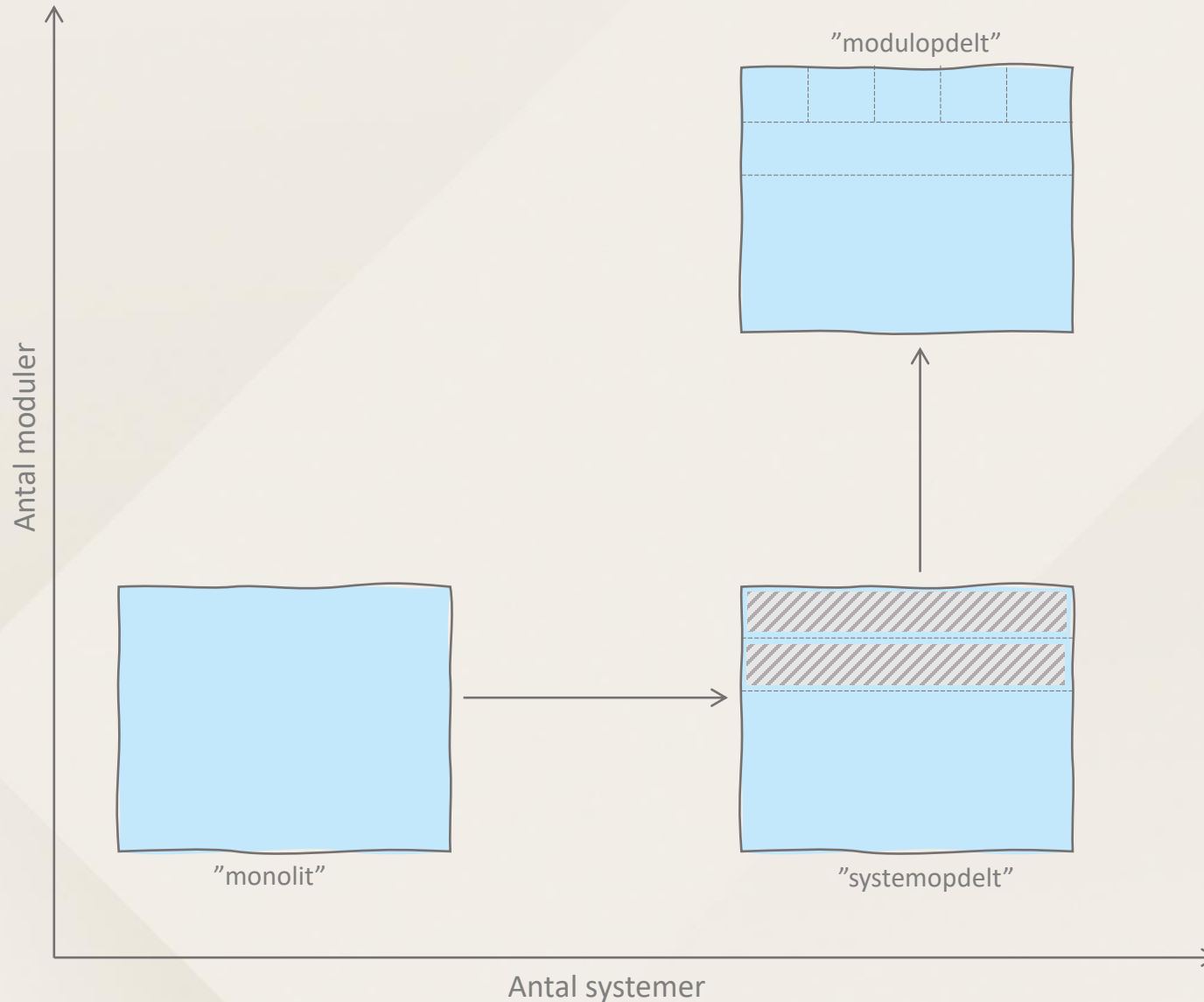
Agil modernisering



Agil modernisering \Rightarrow Kontinuerlig moderinsering



Agil modernisering: Definer og nedbryd moduler



Programplanen er et arbejdsdokument, som er under konstant revurdering



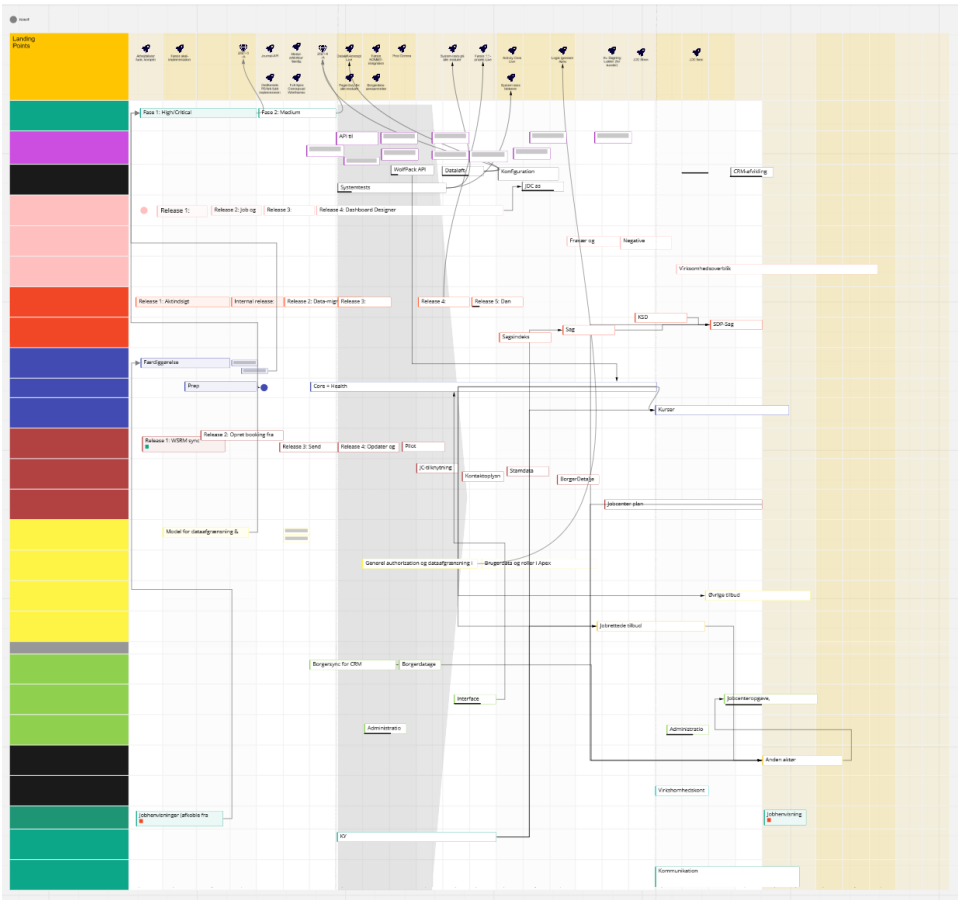
Vi arbejder “fra venstre mod højre” for hele tiden at håndtere de væsentligste afhængigheder og placere opgaverne i de teams, der har de bedste forudsætninger for at lykkes.



Vi arbejder “fra højre mod venstre”, så vi hele tiden kontrollerer, hvad der risikerer ikke at komme med i moderniseringsprojektet, og hvordan vi mitigerer det.



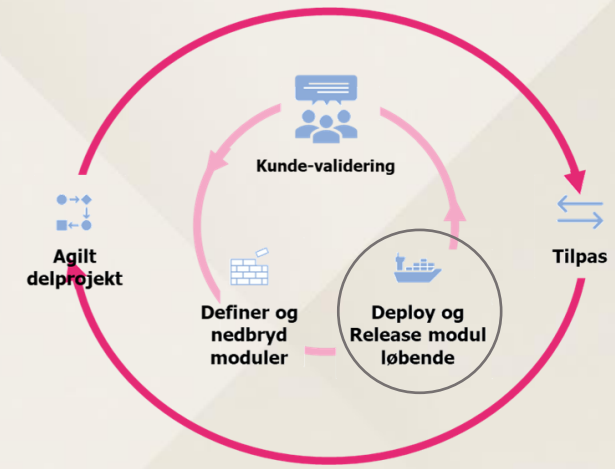
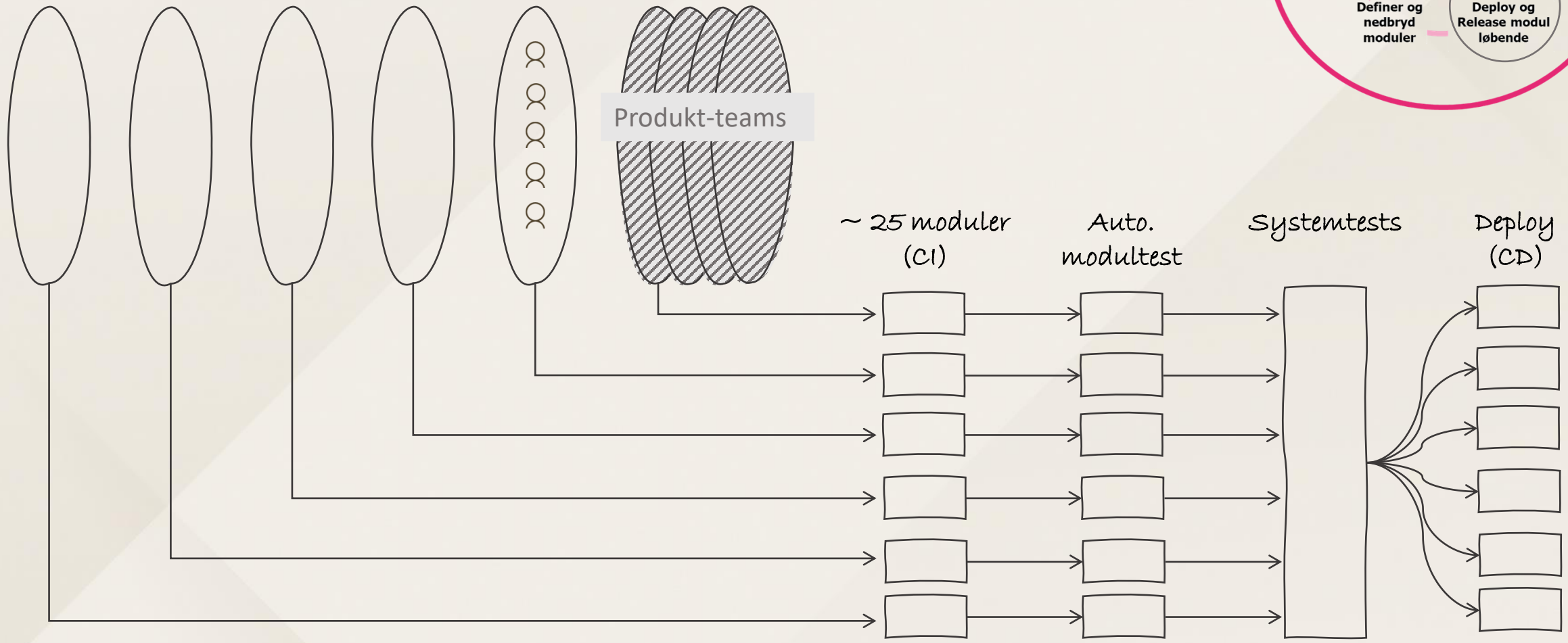
Den viden, der løbende opstår i projekterne, føder ændringer tilbage til den samlede plan.



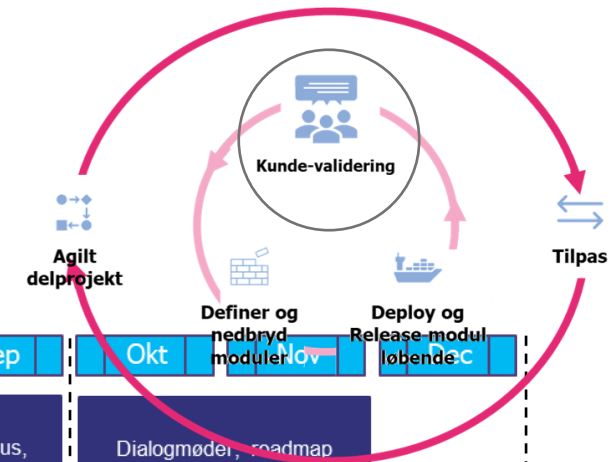
Tilfældigt udsnit af plan (anonymiseret).
Hver række er et del-projekt.
Pile indikerer afhængigheder.
Første rækker indeholder programmets landing points.



5 projektteams

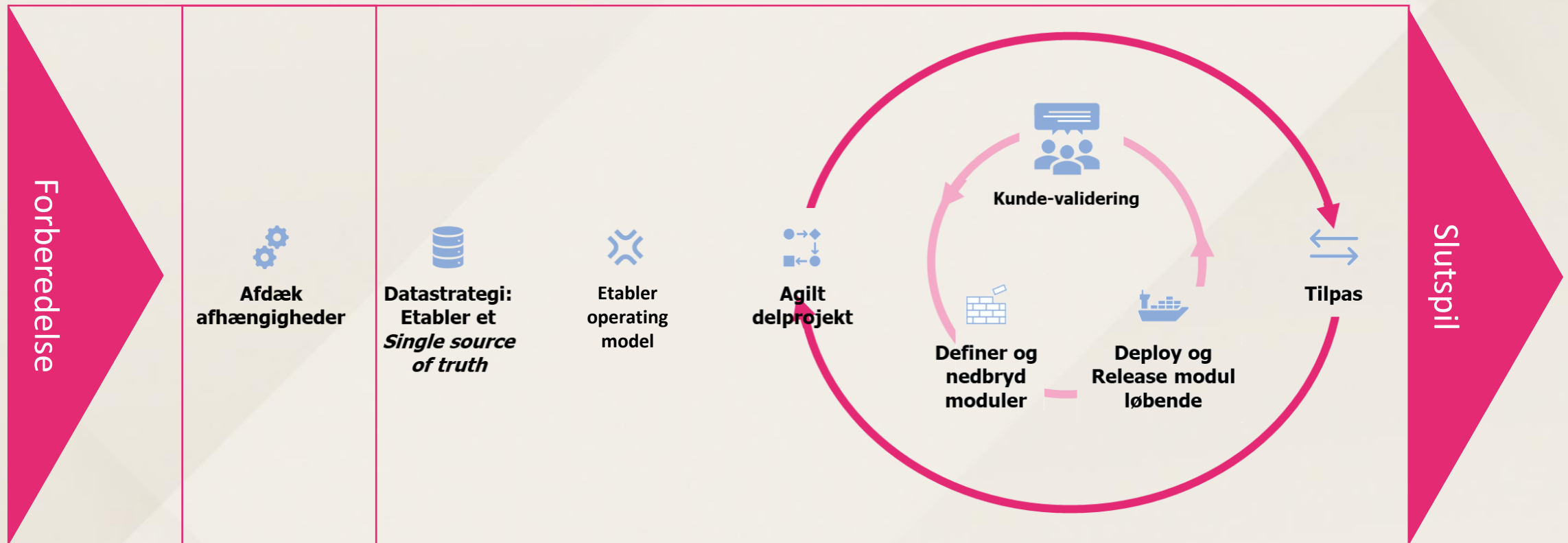


Kundevalidering

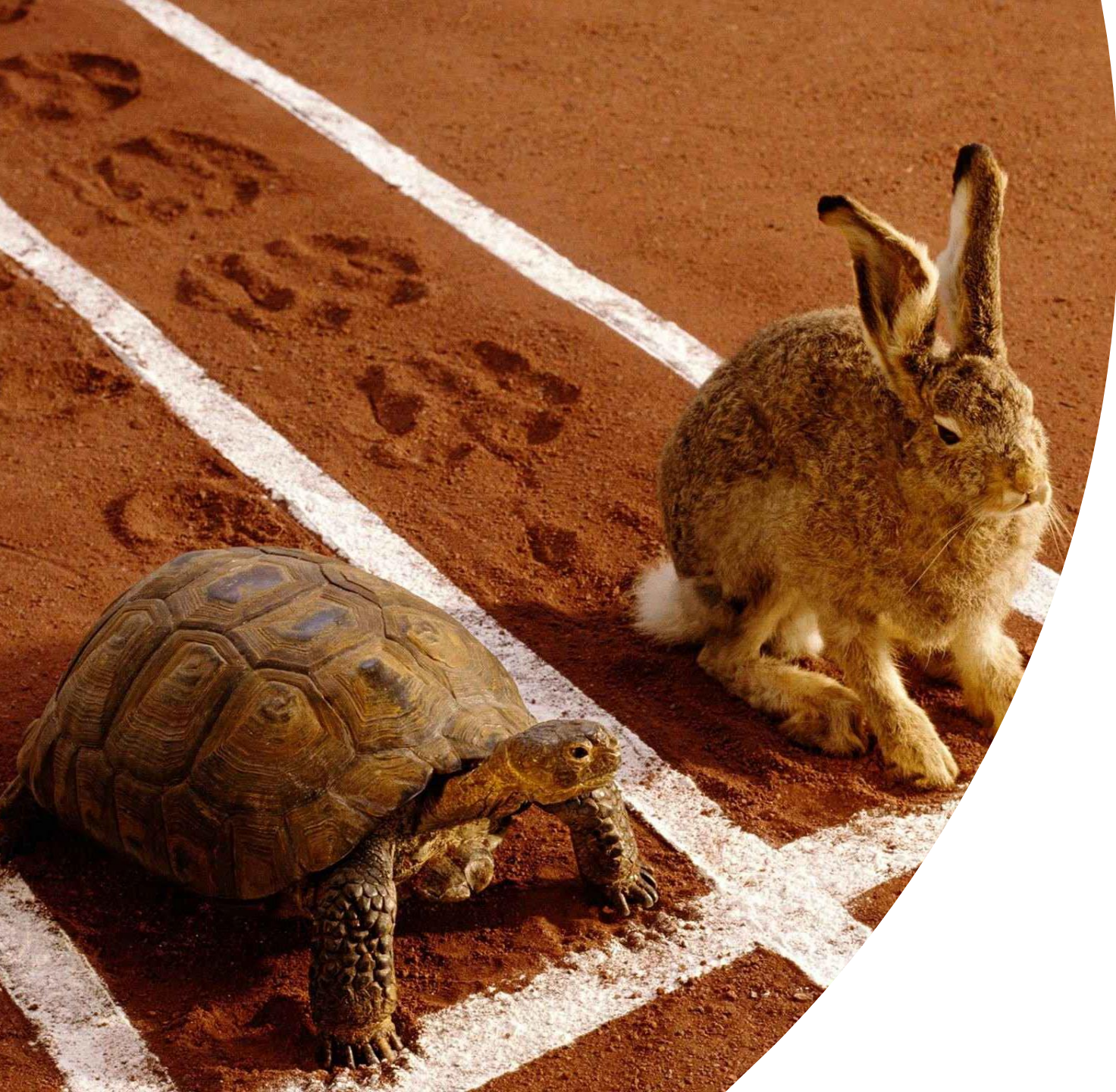


	2020	Jan	Feb	Mar	Apr	Maj	Jun	Jul	Aug	Sep	Okt	Nov	Dec
Dialogmøder	Dialogmøder	Dialogmøder, modernisering i Aarhus, Vejle, Roskilde og Valby							Dialogmøder, modernisering i Aarhus, Vejle, Roskilde og Valby		Dialogmøder, roadmap		
	Brugerforum					Brugerforum						Brugerforum	
Involvering	Delområder	Arbejdsgruppe, arbejdslistor		Arbejdsgruppe, rekruttering & Vitas		Arbejdsgruppe, logs		Arbejdsgruppe, placeringer		Arbejdsgruppe, min plan			
	Delområder	Arbejdsgruppe, a-kasse		Arbejdsgruppe, borgerkommunikation		Arbejdsgruppe, Samtaler		Arbejdsgruppe, visitation		Arbejdsgruppe, Jobsøgning & CV			
Kommunikation	Webinarer	1-2 webinarer pr. måned			1-2 webinarer pr. måned			1-2 webinarer pr. måned		1-2 webinarer pr. måned			
	Nyhedsbreve	10	11		12	13		14	15		16	17	
	Kundeweb/FAQ		135	136		137	138		139	140		140	141
Governance	Forankring og styring	Styregruppe			Styregruppe			Styregruppe		Styregruppe			
		Status og forankring af moderniseringen i Advisory Board			Status og forankring af moderniseringen i Advisory Board			Status og forankring af moderniseringen i Advisory Board		Status og forankring af moderniseringen i Advisory Board			

Agil moderniseringsmodel



Applikation og data modernisering med Cloud

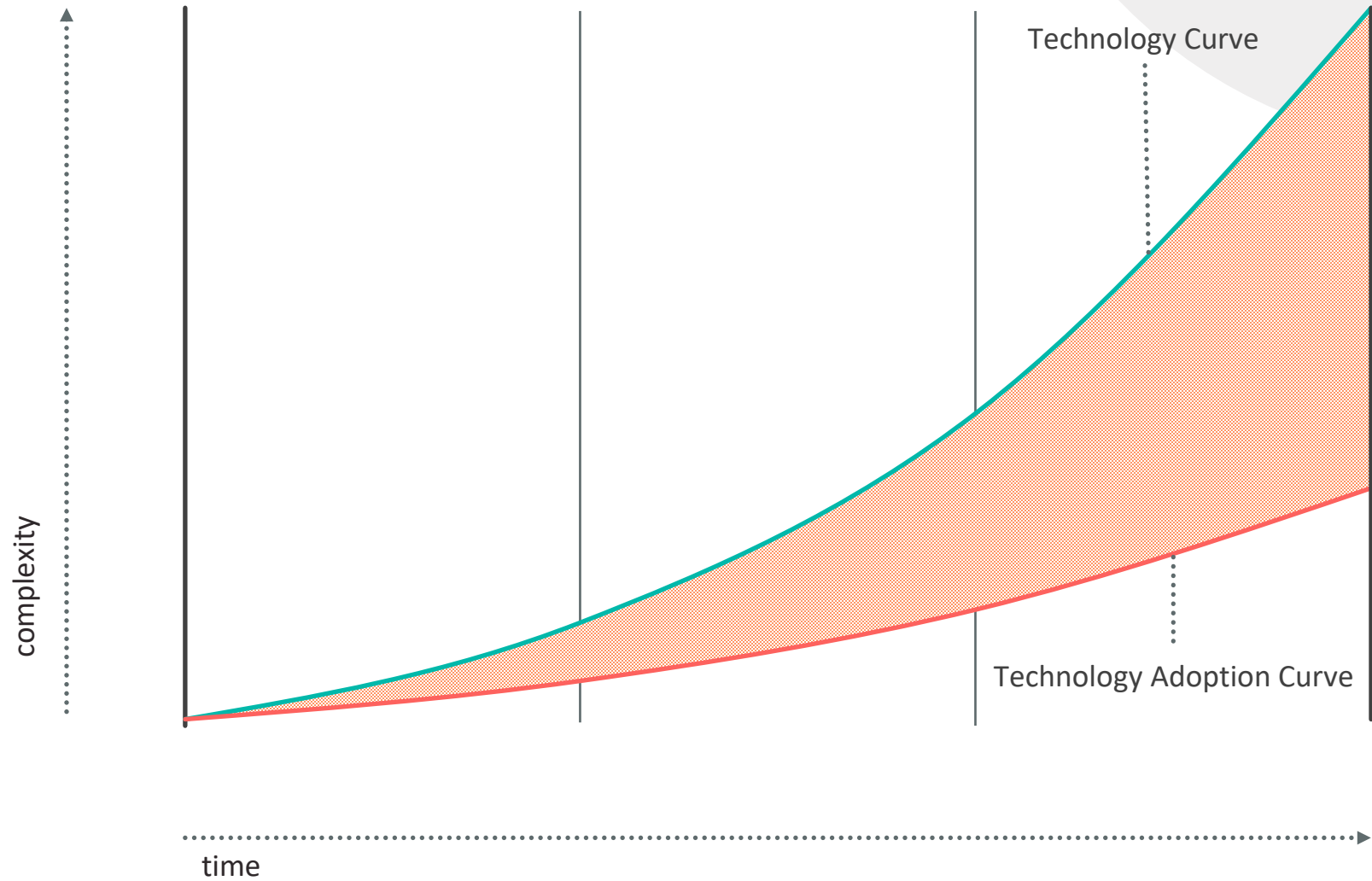


The Need for Speed & Innovation

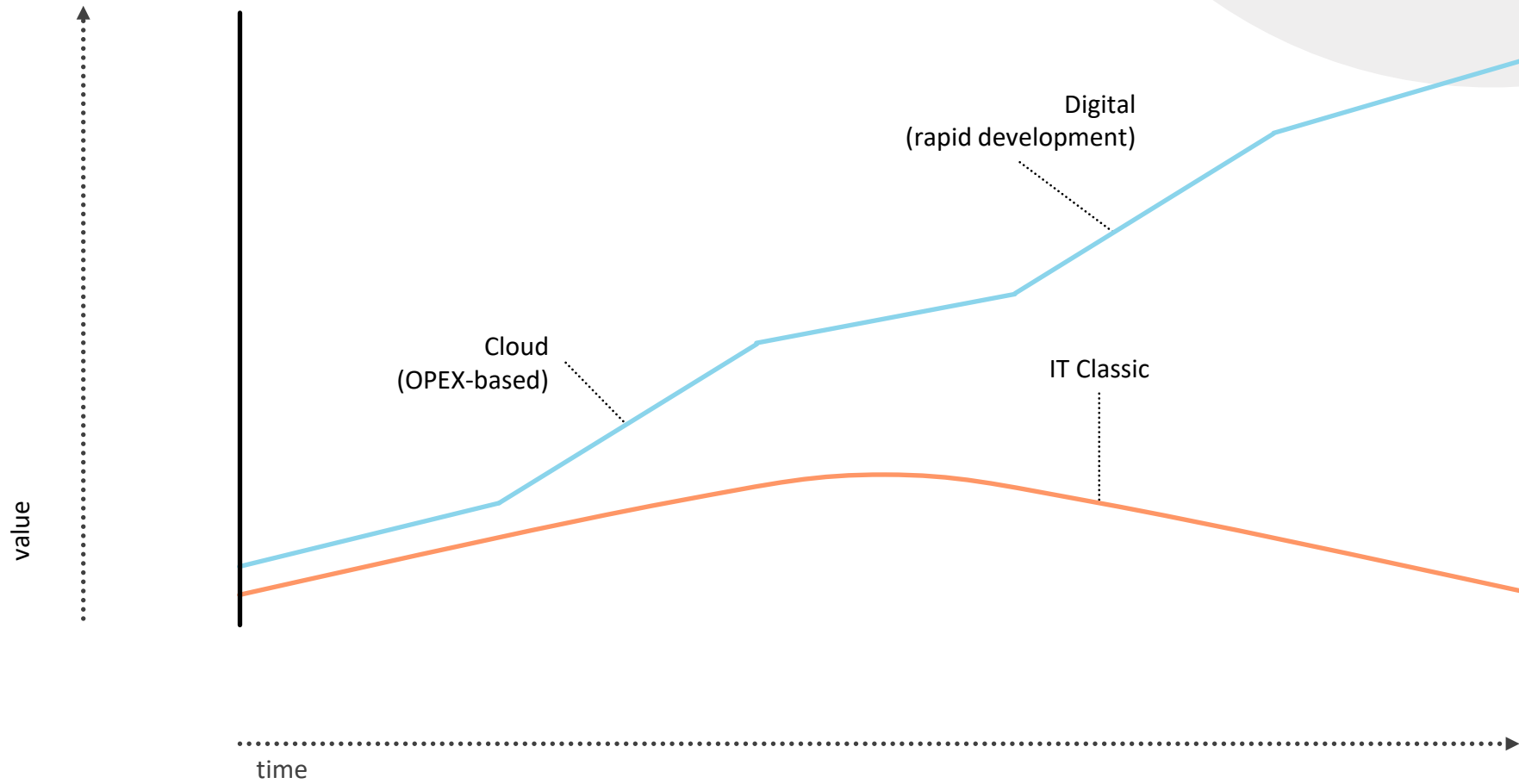


Requires
a new approach
and new tools

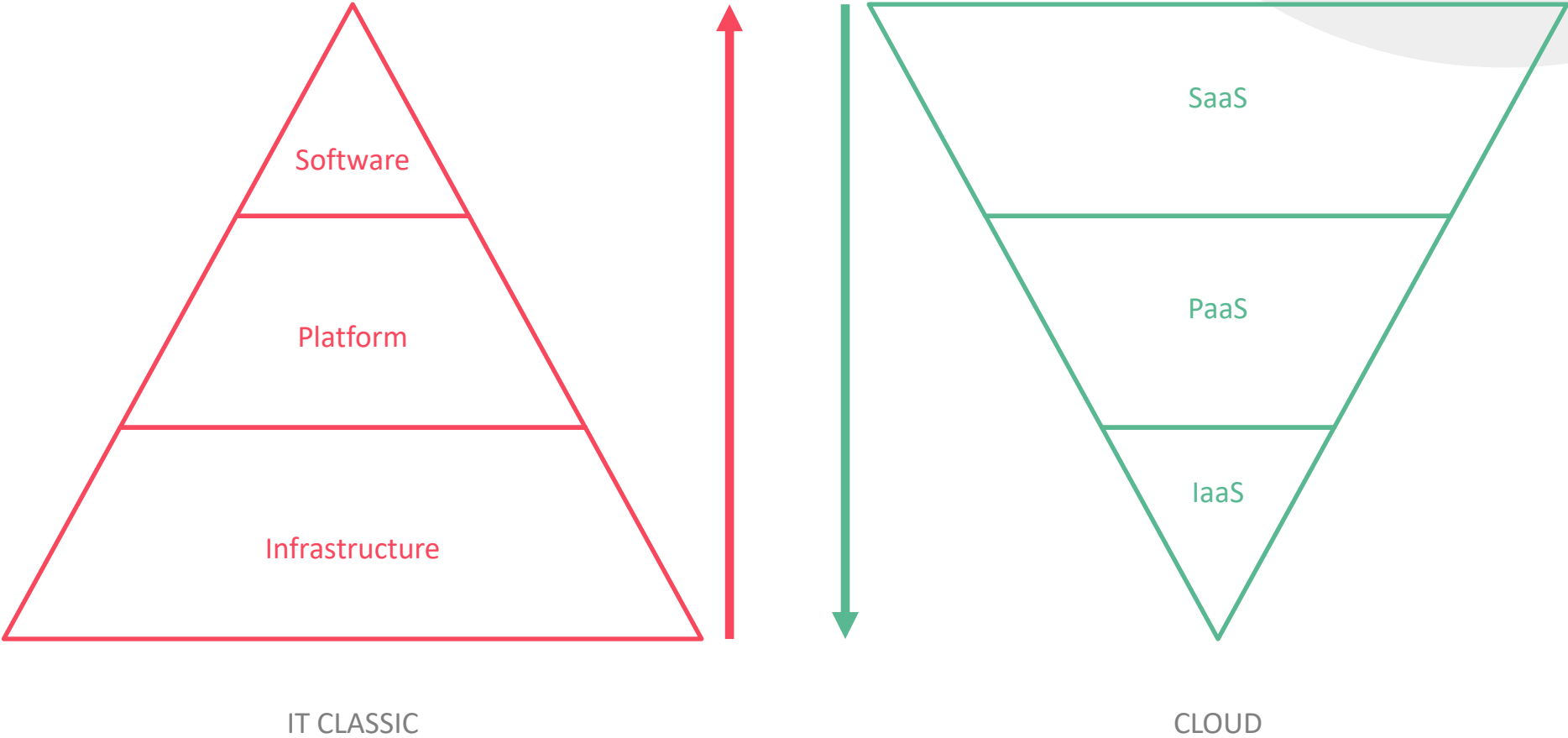
What is technical debt ?



In Debt ? Change the Game



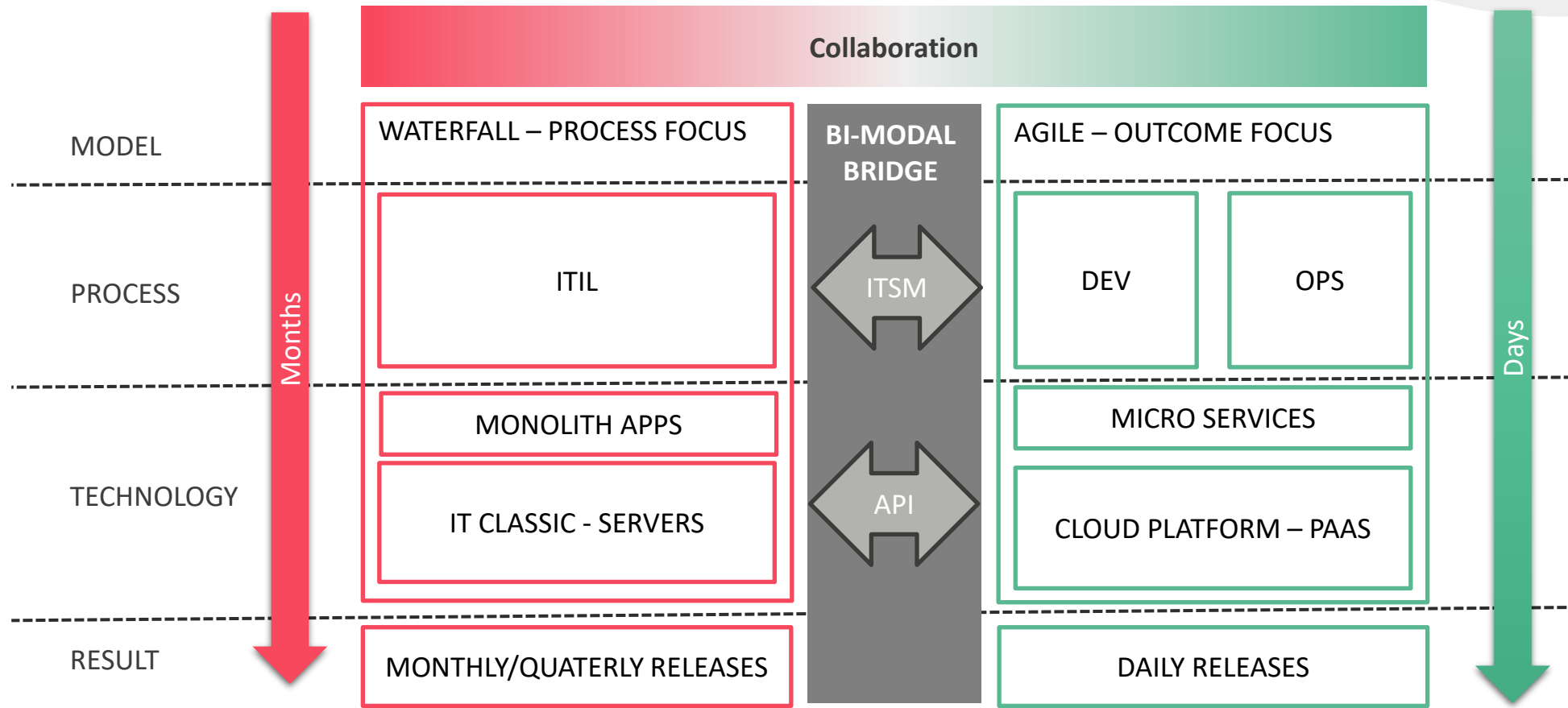
Cloud First



Establish a new Operating Model



Digital Ideas

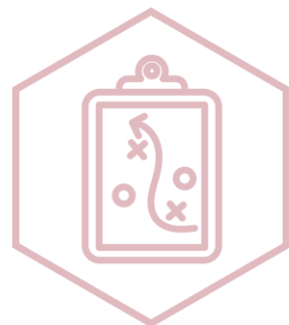


Application Modernization Options



RE-HOST (Lift & Shift)

Often referred to as a lift-and-shift migration, this option doesn't require code changes. You can use it to migrate your existing applications to Azure quickly. Each application is migrated as is to reap the benefits of the cloud without the risk and cost associated with code changes.



RE-FACTOR (clean up a little)

Often referred to as a "repackaging," refactoring requires minimal changes to applications so that they can connect to Azure platform as a service (PaaS) and use cloud offerings.

For example, you could migrate existing applications to Azure App Service or Azure Kubernetes Service (AKS).



RE-ARCHITECT (clean up a lot)

Rearchitecting for migration focuses on modifying and extending application functionality and the code base to optimize the application architecture for cloud scalability.

For example, you could break down a monolithic application into a group of microservices that work together and scale easily

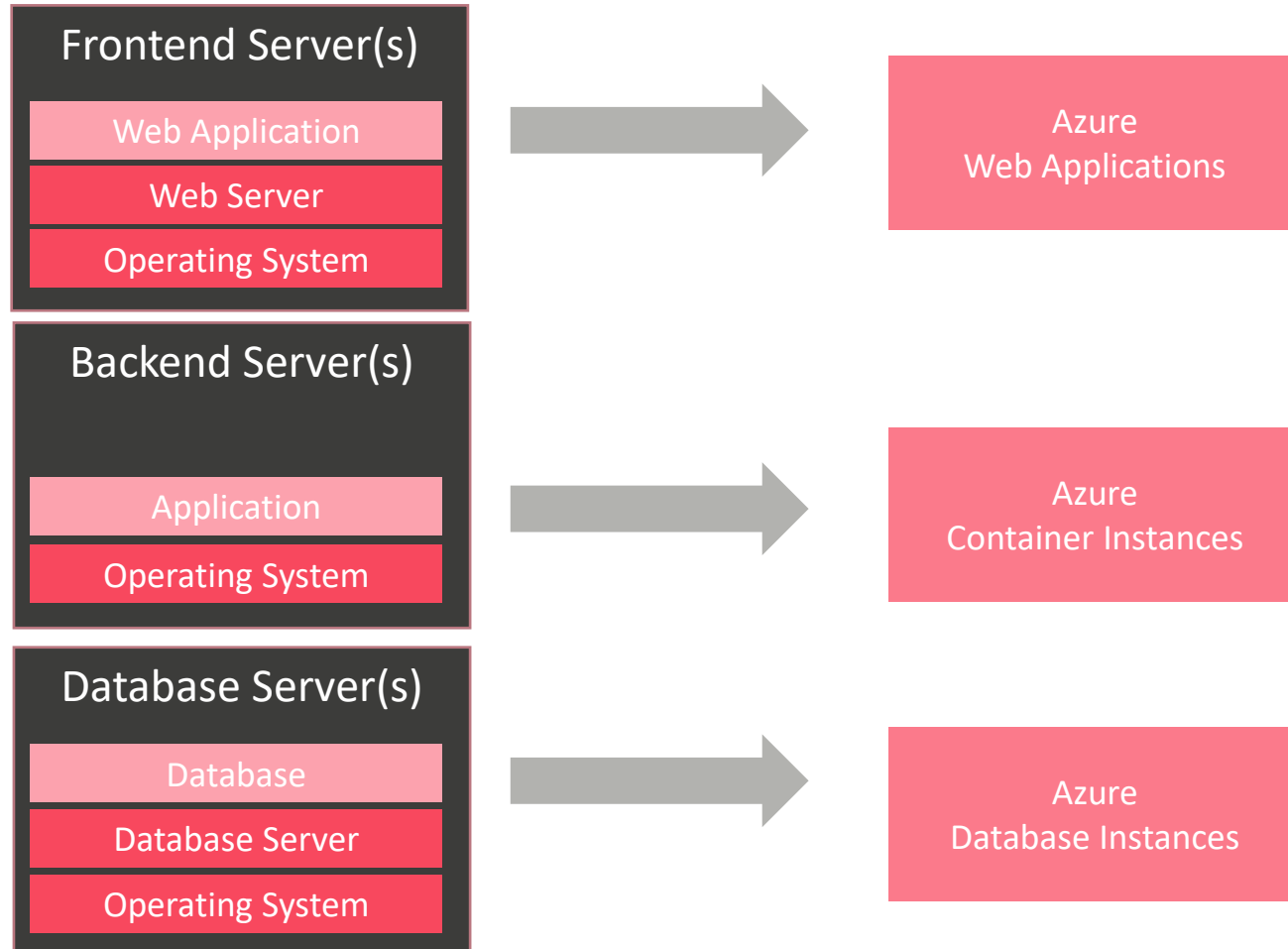


RE-BUILD (start from scratch)

Rebuild takes things a step further by rebuilding an application from scratch using Azure cloud technologies.

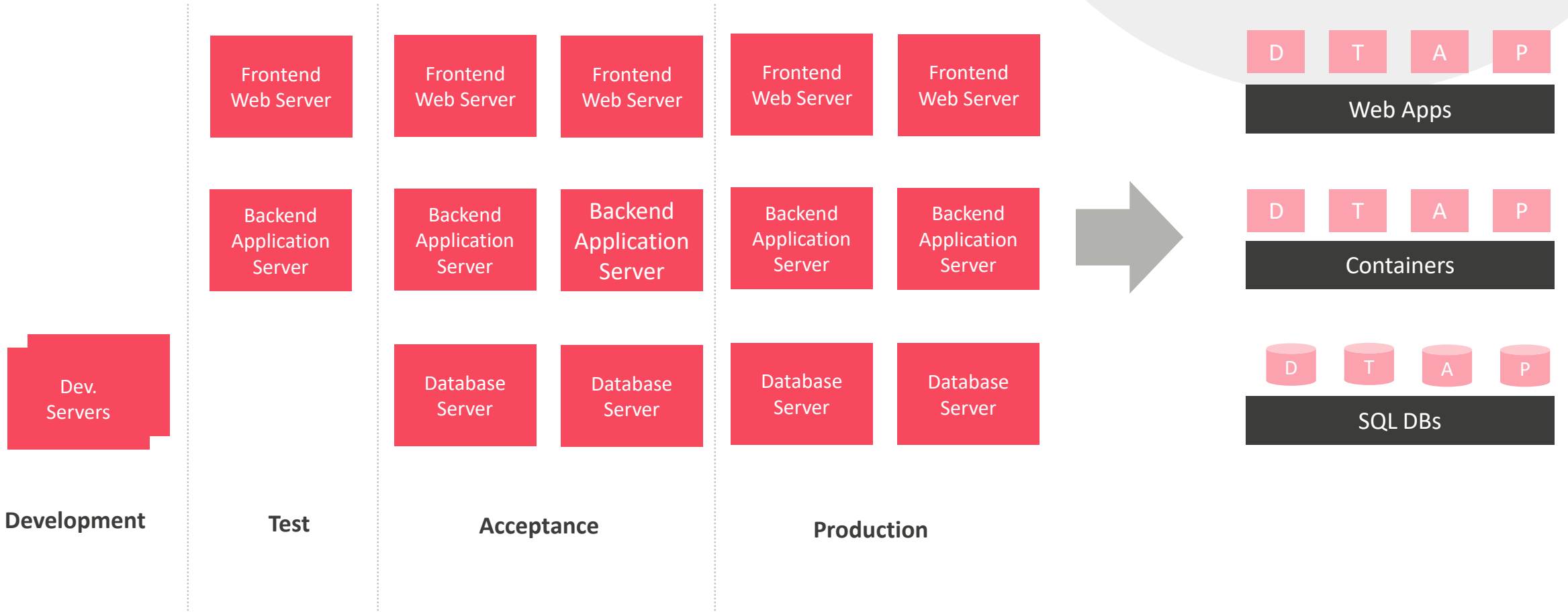
For example, you could build greenfield applications with cloud-native technologies like Azure Functions, AI, SQL Managed Instance, and Azure Cosmos DB.

Application Re-Factor Modernization Example

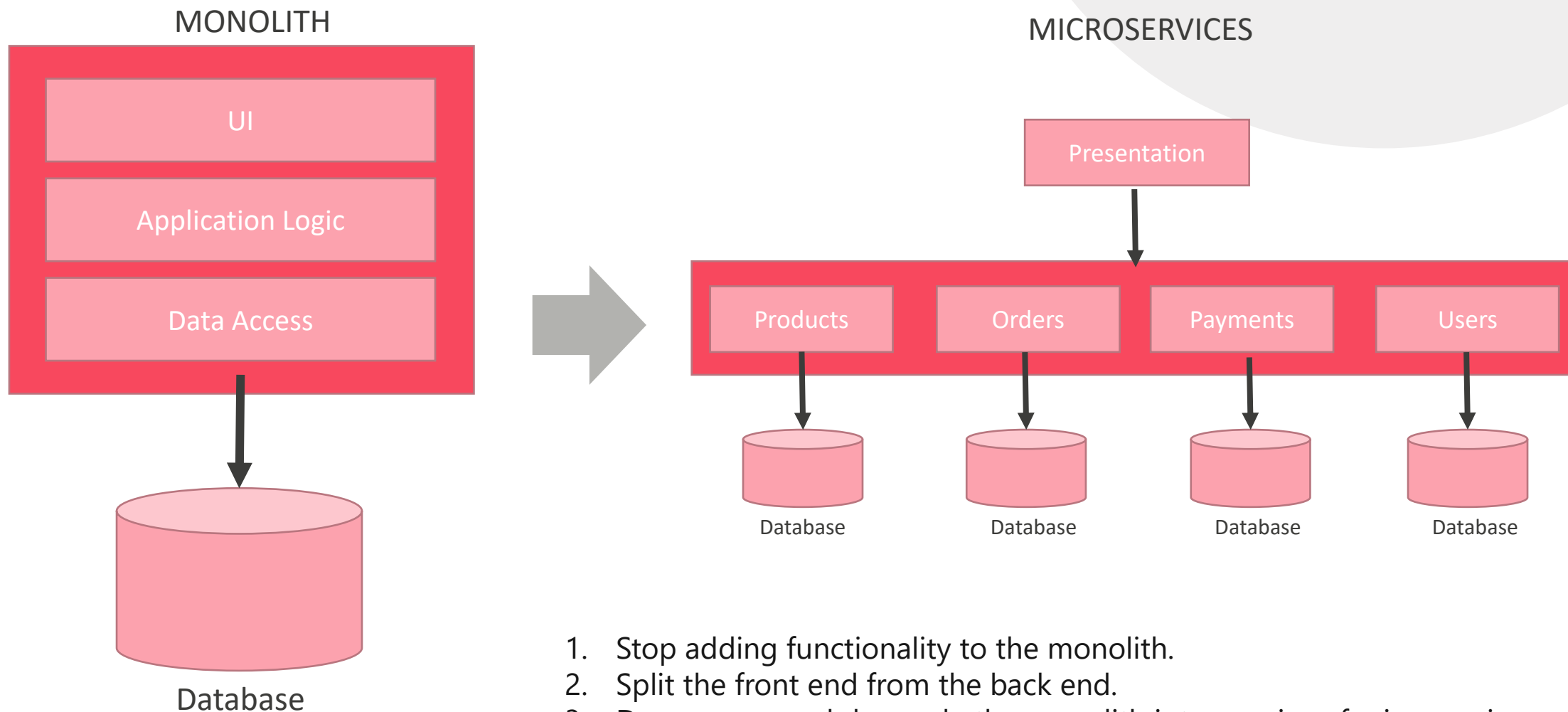


Application Re-factor Example

App. Re-factor

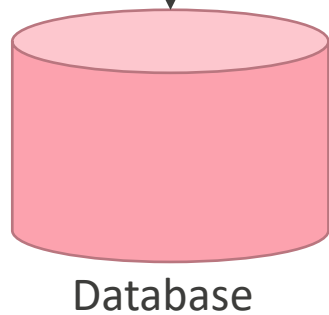
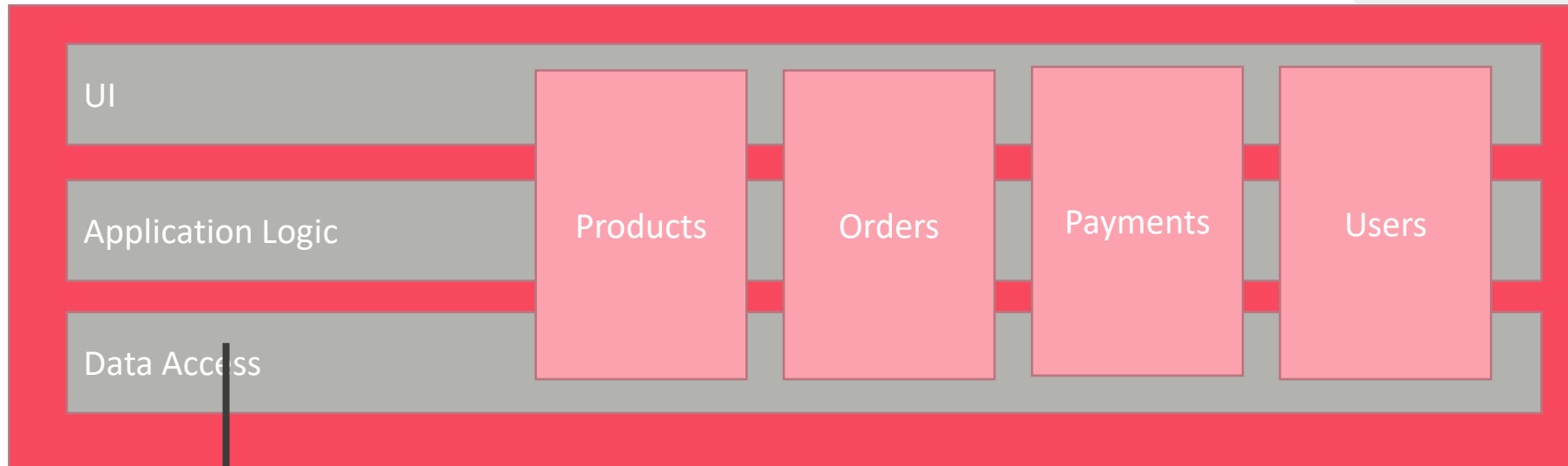


Re-architect Monolith to Microservice



1. Stop adding functionality to the monolith.
2. Split the front end from the back end.
3. Decompose and decouple the monolith into a series of microservices.

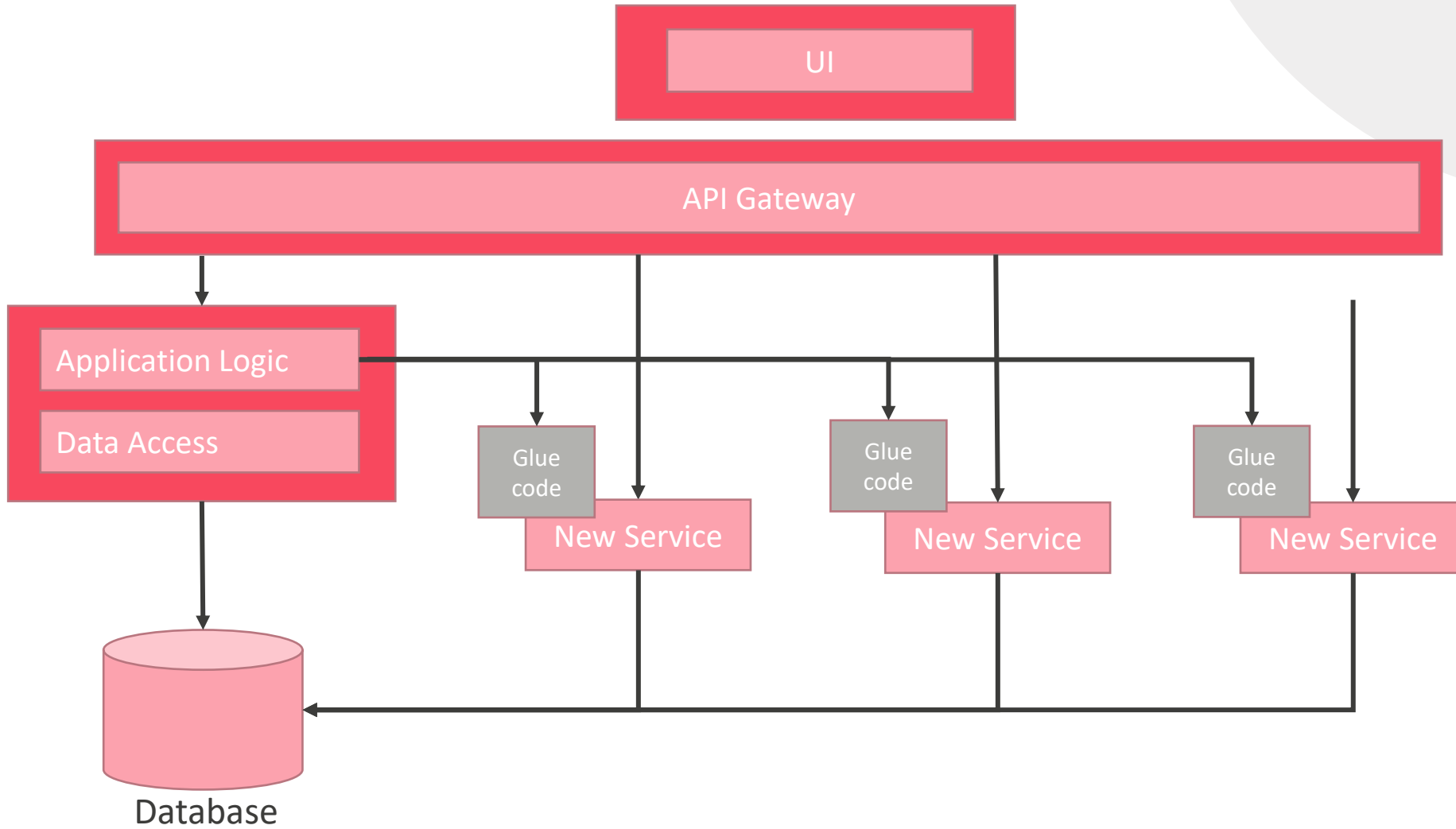
Domain Driven Design



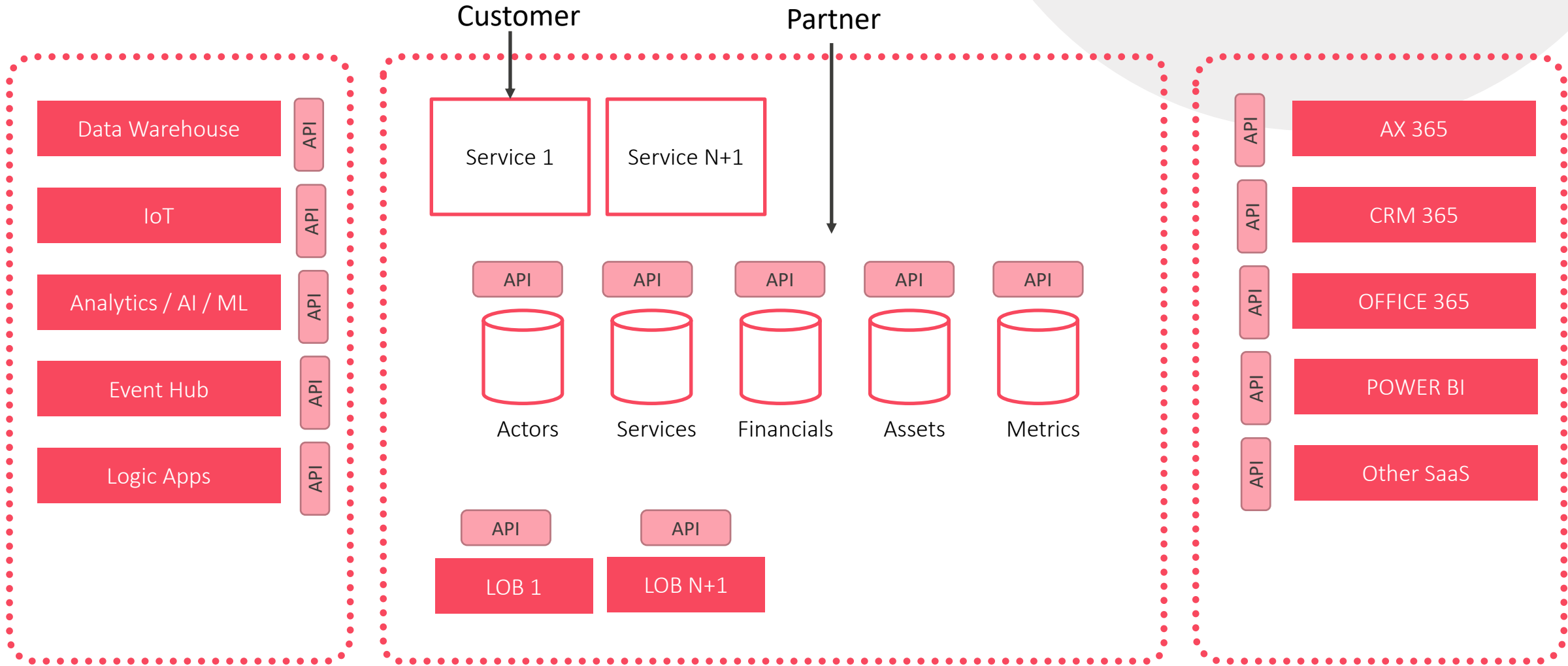
1. Start with a *ubiquitous language*, a common vocabulary
2. Identify the relevant modules in the monolithic application and apply the common vocabulary to them.
3. Define the domain models of the monolithic application. The domain model is an abstract model of the business domain.
4. Define *bounded contexts* for the models. A bounded context is the boundary within a domain where a particular domain model applies. Apply explicit boundaries with clearly defined models and responsibilities.

The bounded contexts identified in step 4 are candidates for refactoring into smaller Microservices

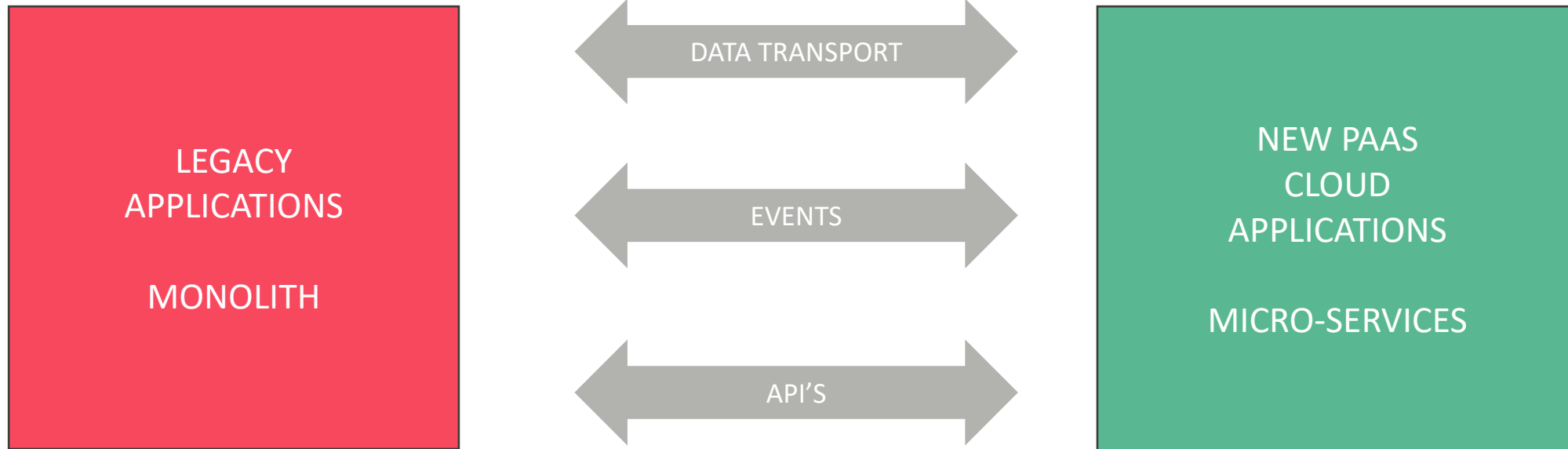
Retire Monolith



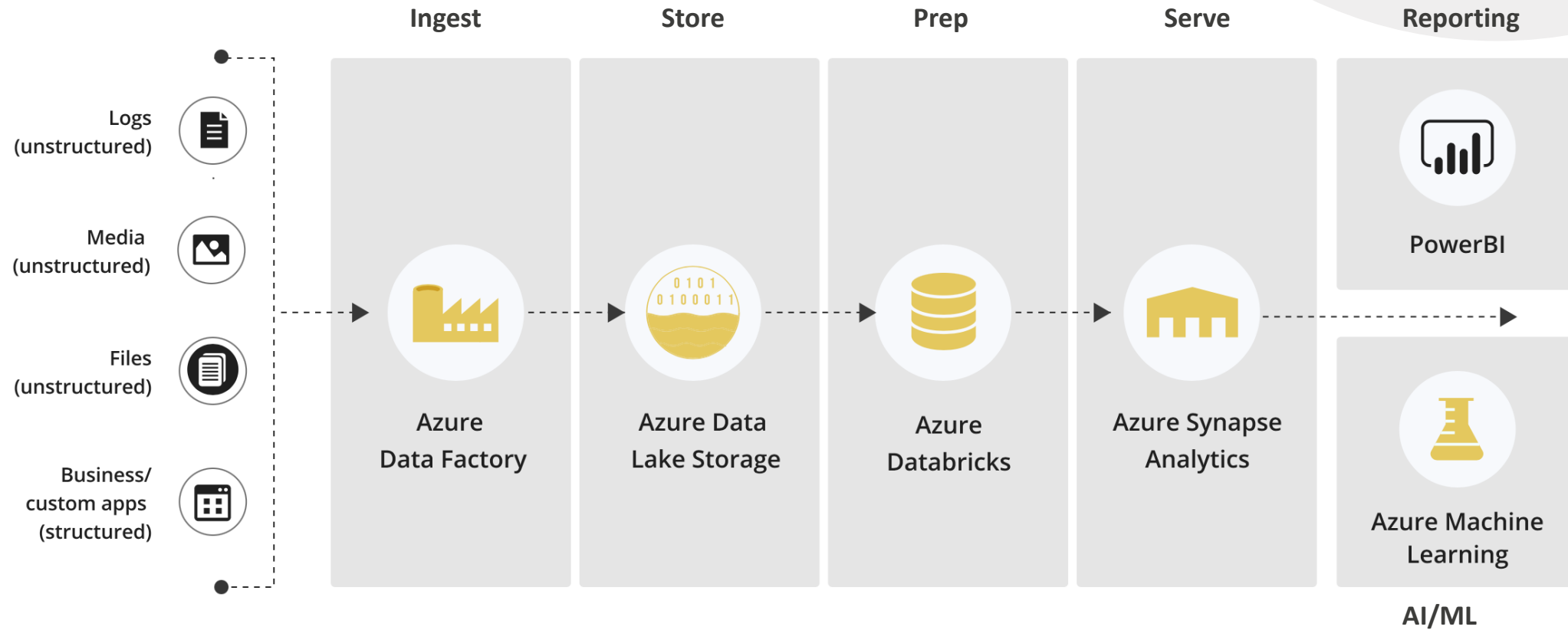
Revolution (Greenfield) vs. Evolution (Brownfield)



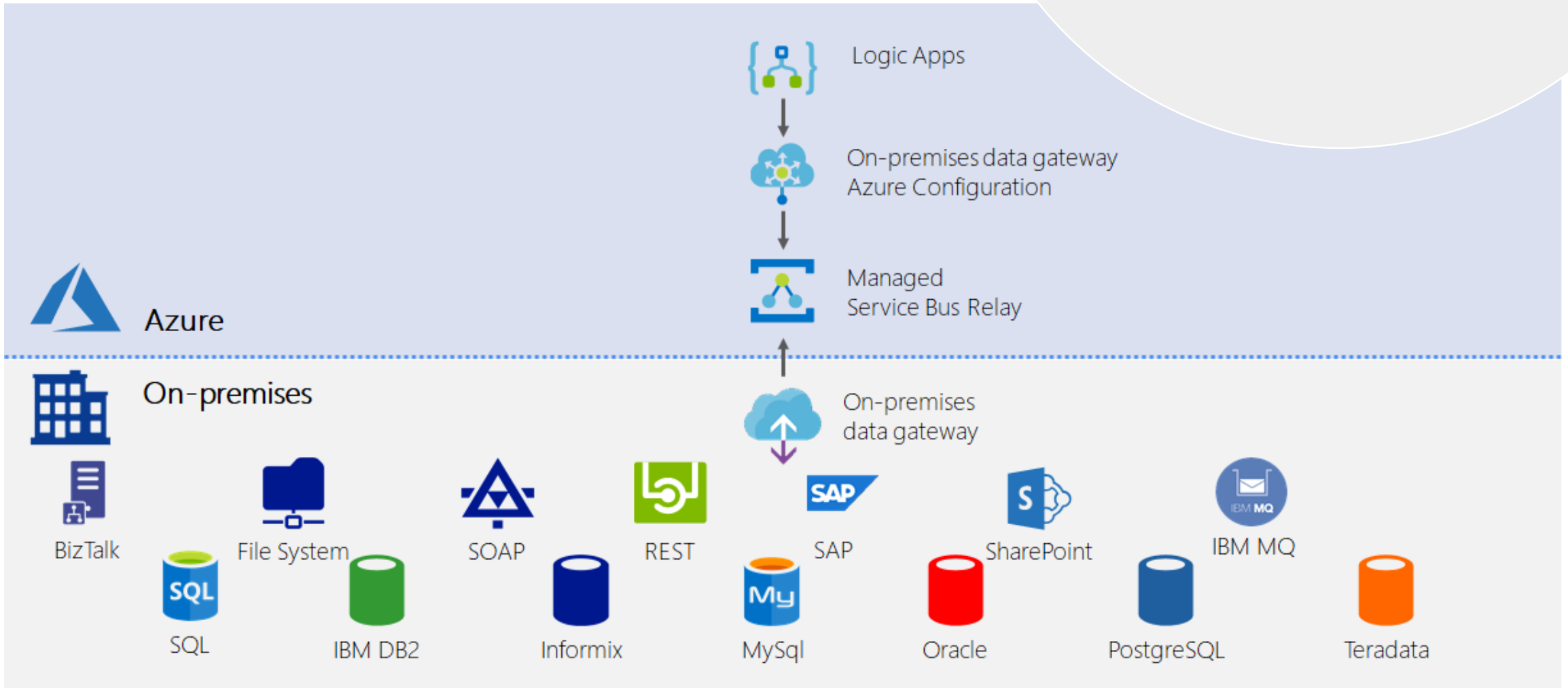
Co-existence challenges



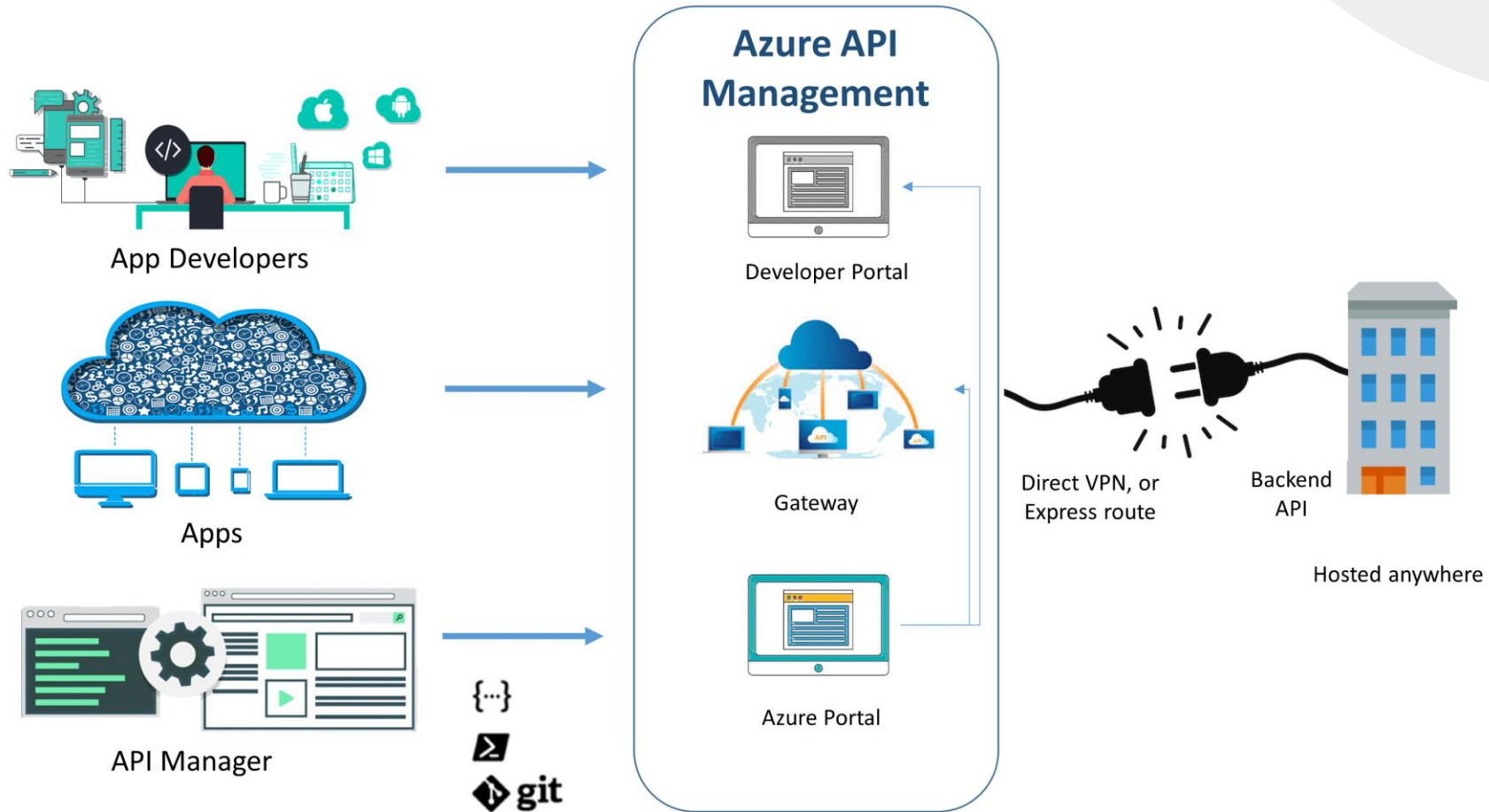
Data Platform - Data Transformation, Analytics & AI



Events, Workflows and Hybrid Service bus



API Management (Azure API-M)





Thank you.



NYSGERRIG PÅ MERE?

VI STÅR TIL RÅDIGHED

Kristian Kossmann

Salgschef, Schultz

kko@schultz.dk

+45 4195 4791

Simon Svarrer

Adm. direktør, Schultz

ssv@schultz.dk

+45 4195 4737

Morten Maate

Udviklingschef, Schultz

mma@schultz.dk

+45 4195 4762



devoteam



CLOUDEON

Part of Devoteam

Peter Nørregaard

Expert Director, Devoteam

peter.noerregaard@devoteam.com

+45 4178 1316

Frank Mogensen

CTO, CLOUDEON – Part of Devoteam

fmo@cloudeon.com

+45 2133 1133

VED VEJS ENDE
TAK FOR I DAG

