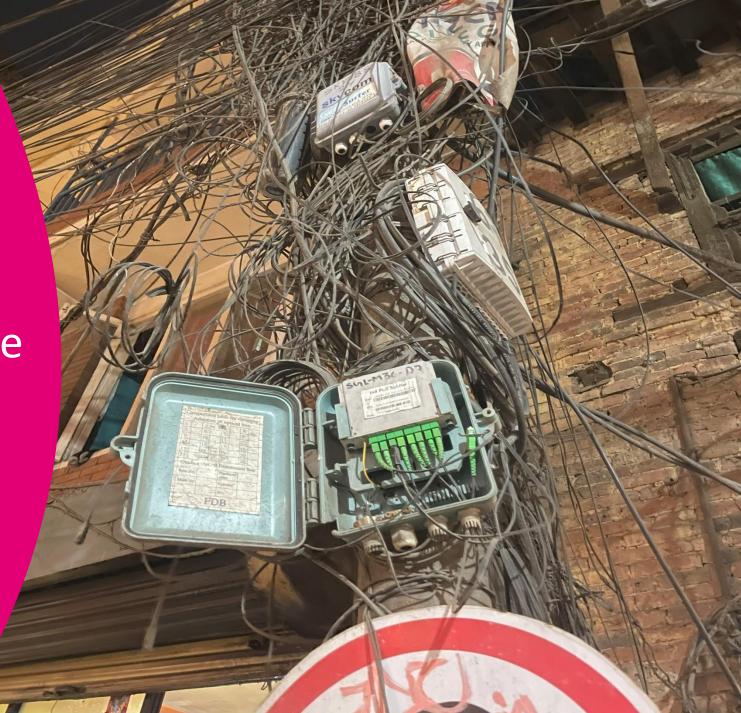
Scaling data pipelines @Telekom

Dr. Georg Heiler

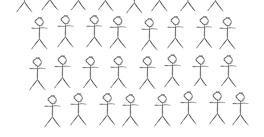
开



## There is a chaos out there

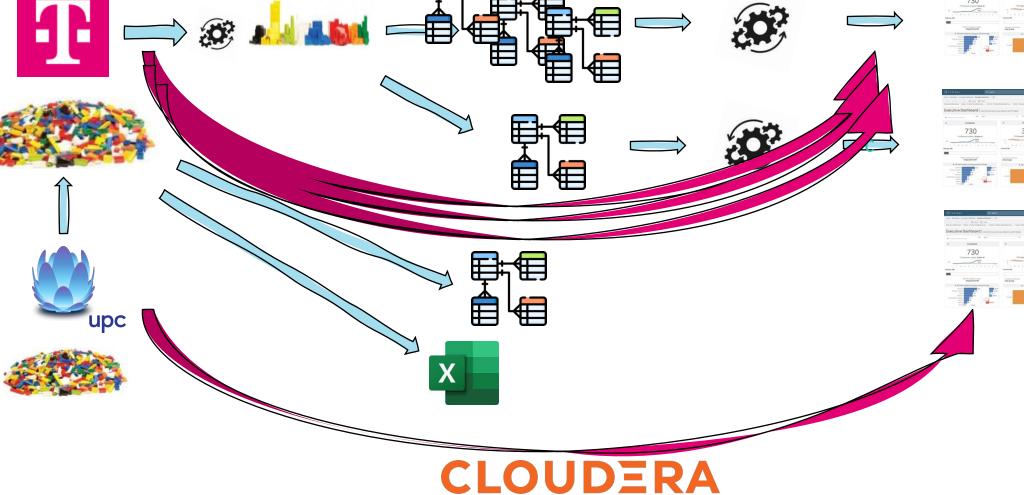












## How did we end here? Time!

business grows (merger) demand for data grows methodology and tooling changes

- Missing lineage
- Missing semantics
- Missing collaboration
- High lead times
- Limited quality

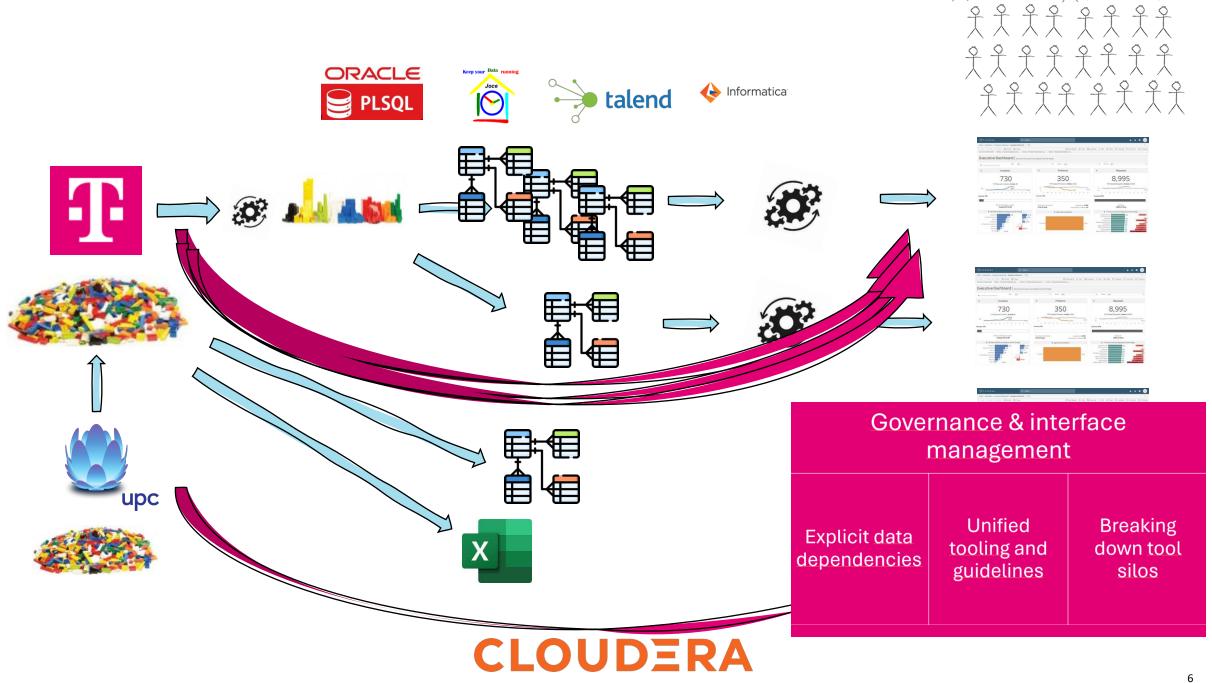
"[...] model behavior is not determined by architecture, hyperparameters, or optimizer choices. It's determined by your dataset, nothing else."

James Betker Research Engineer, Open AI <u>https://www.youtube.com/watch?v=lvhtTu9CTAU</u>

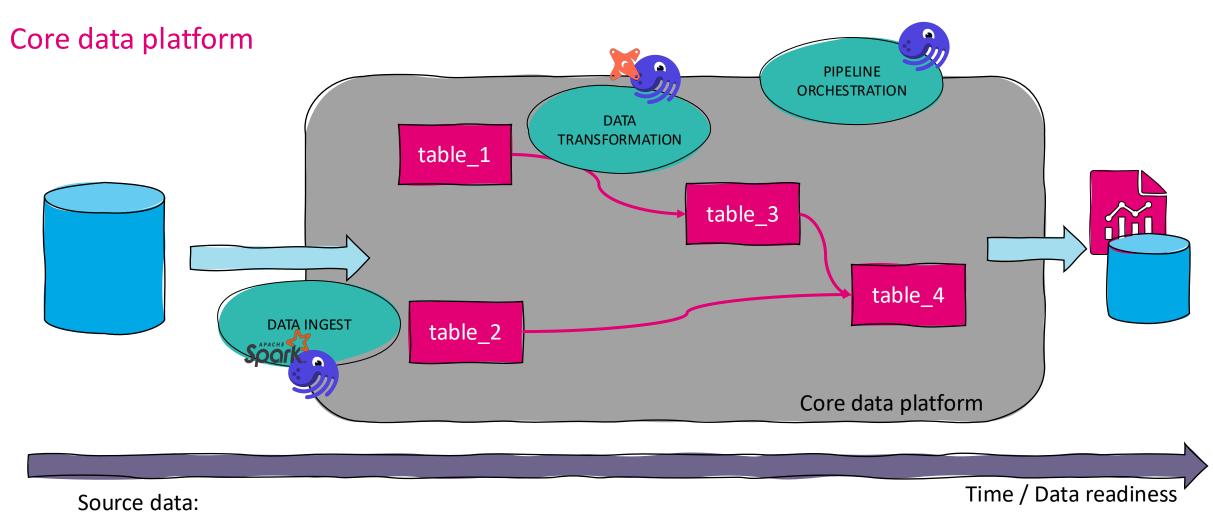
## Governance & interface management

# Explicit data dependencies

Unified tooling and guidelines Breaking down tool silos

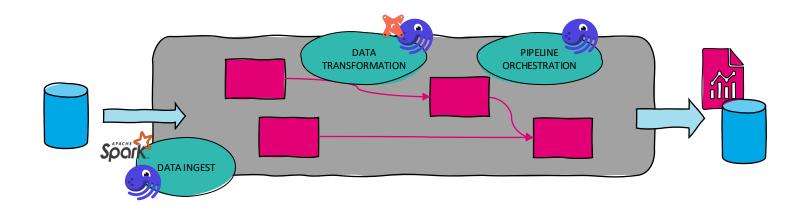


 $\land \land \land \land \land \land \land \land$ 

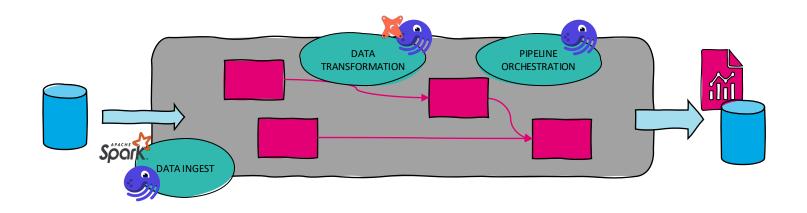


- Kafka
- Files
- Database systems

# I fear that what we build is very hard to push into the business units

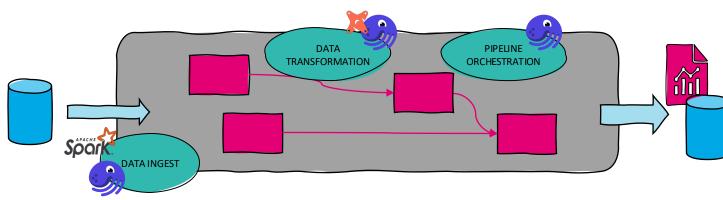


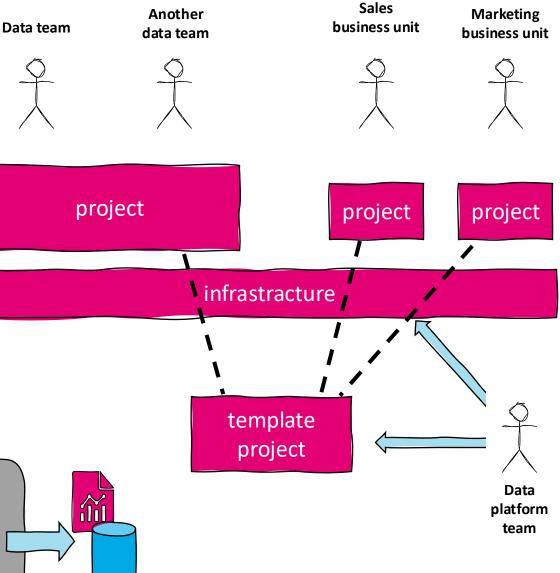
## know I fear-that what we build is very hard to push into the business units



#### Observation

- Process is straight forward: ingest, transform, use
- Everything we do we do for business to provide better service
- Hard to scale across company
- Dividing people into develop framework and use framework groups
- Thinking in a **building block** structure
- Introduce modern tooling supporting software engineering practices: dbt, dagster, pixi, docker
- Introduce new processes, modeling and metadata tooling for better governance





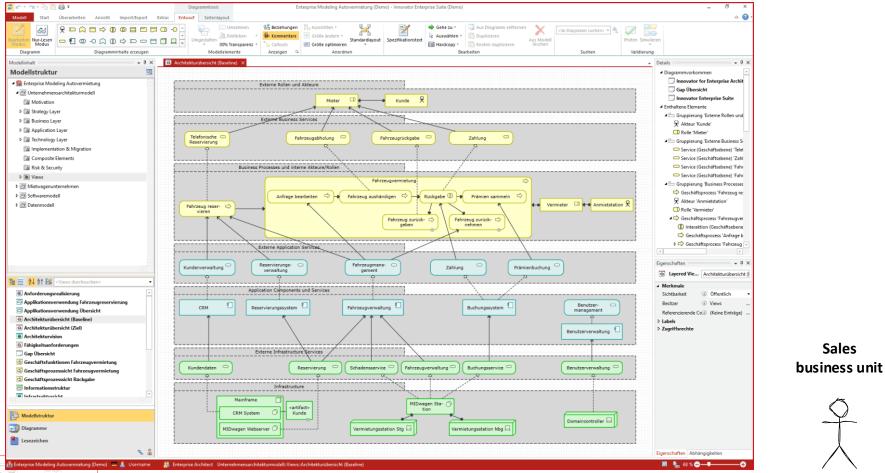
...

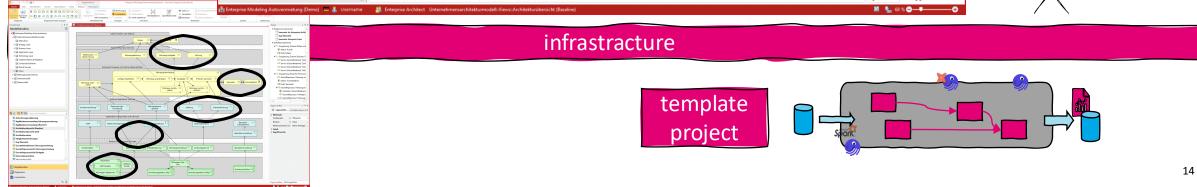
Having a data platform team doesn't mean that your platform scales

- 1. Building block with governance, modeling and software-engineering principles
- 2. Understanding data platform vendor war
- 3. How to bridge department and tool silos



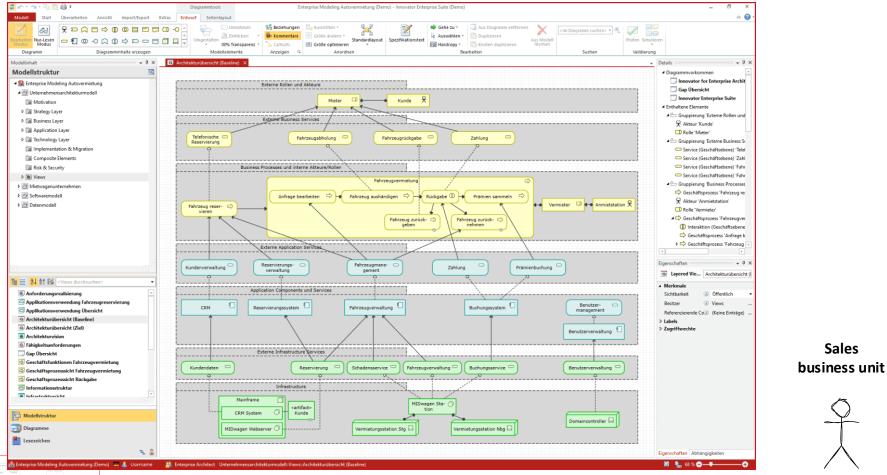


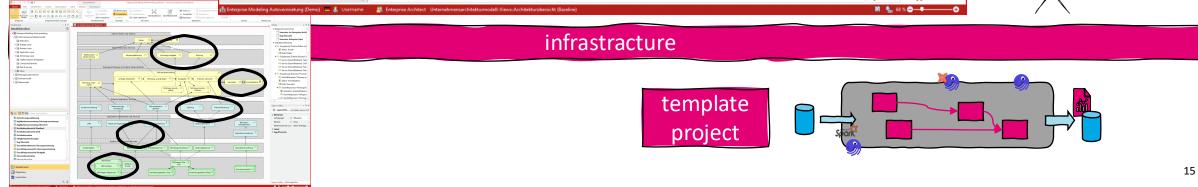


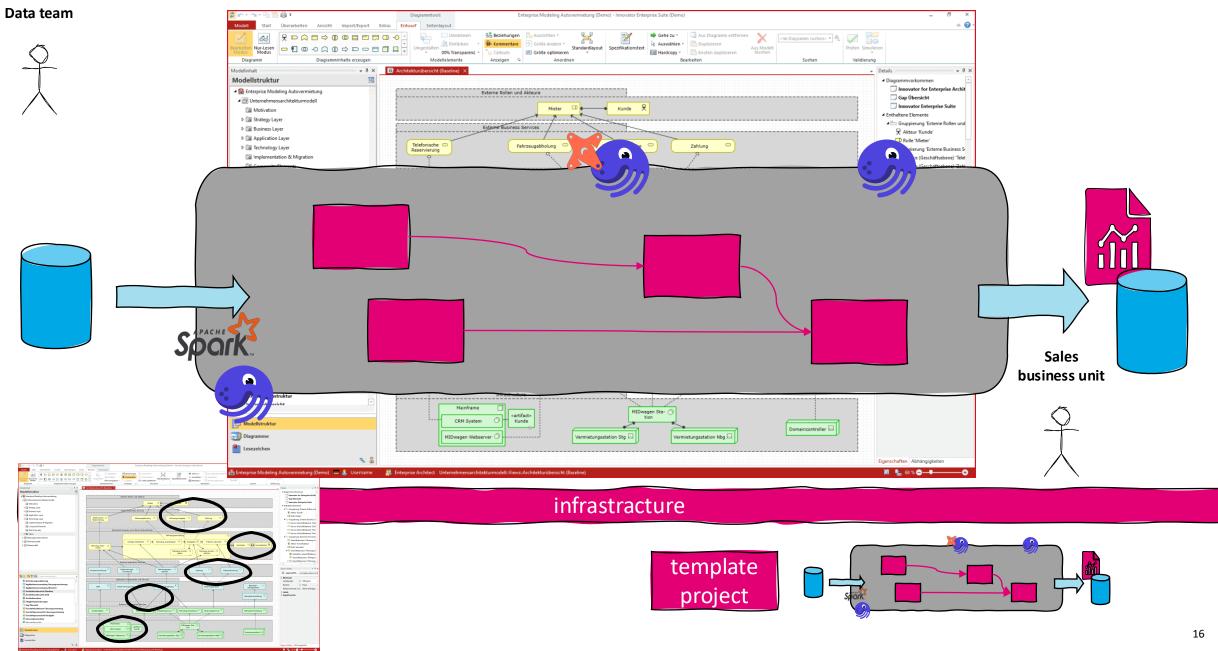


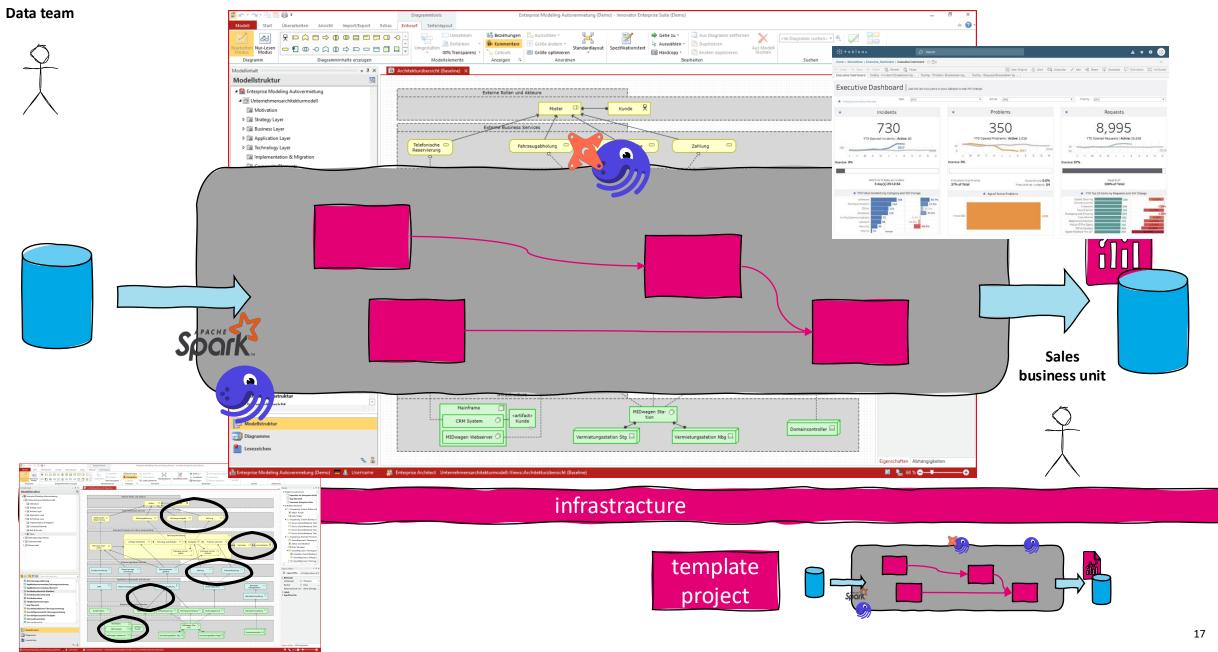


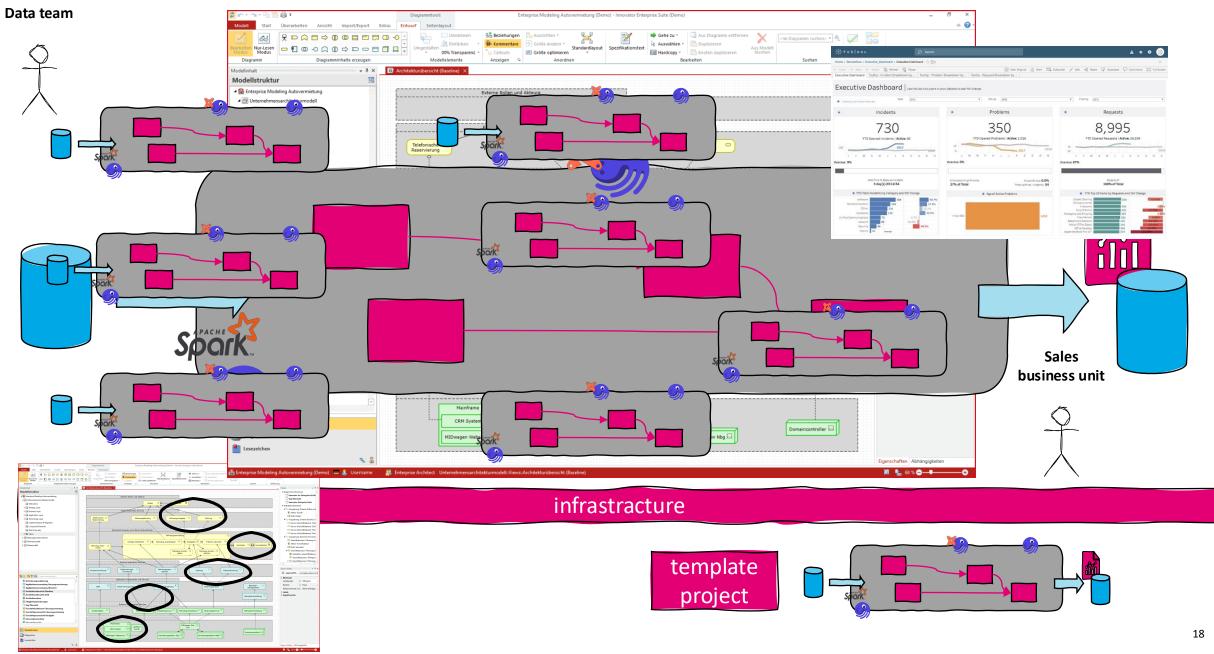


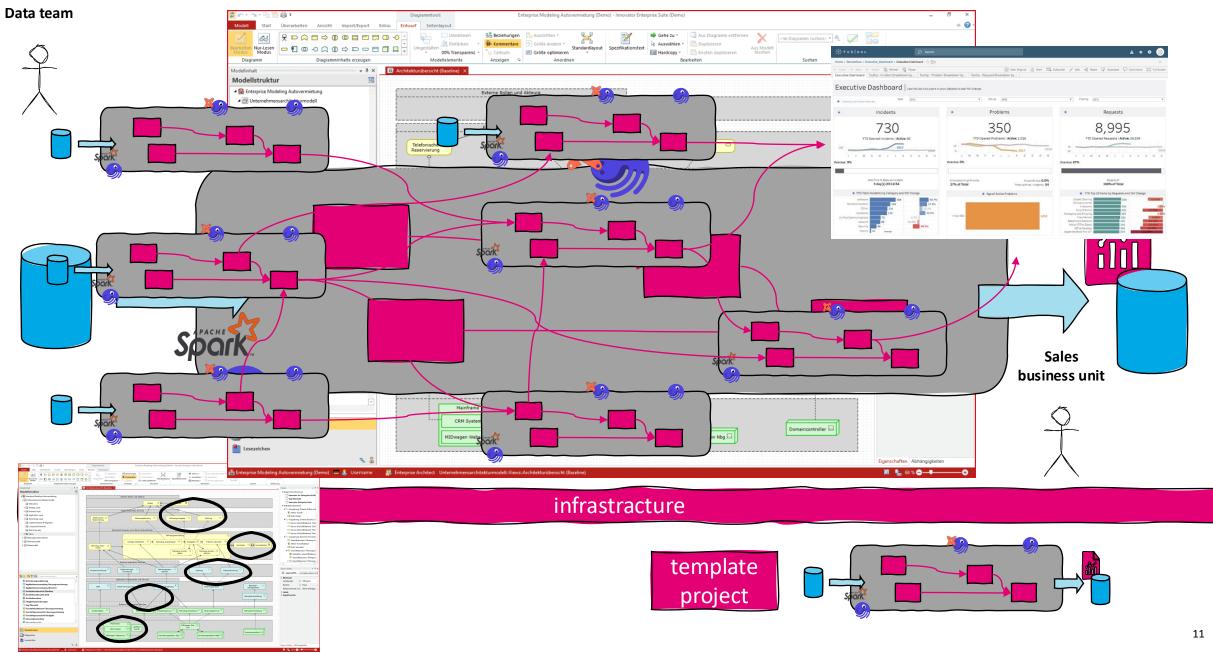










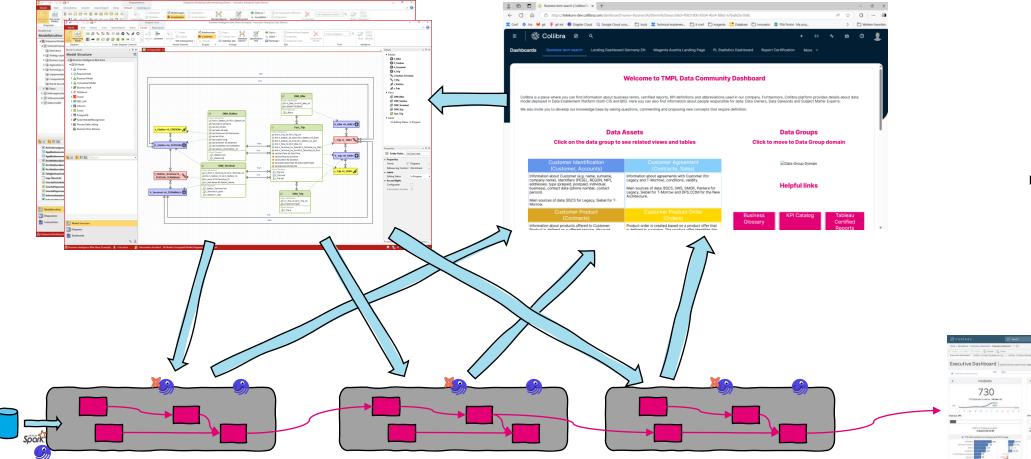


## Multi-project setup is a challenge!

- 1. Dedicated team maintaning infrastructure and template project
- 2. Governance and modeling tools

#### Modeling and governance are keys for success

#### **Development phase**



me, you

Contract and set of the set of th

infrastructure

**Exploration phase** 

#### Understanding data platforn

	10000	T LANDALAKES / -		DATA WANNER		NUMBER	_	
	Ofering Intellin	(dremio	distantia de la	awster (			athen the fermitian	in the state of the
IBM Own	VERTON	1 100 100 to		Dimensel				the second of the
		011-04		ORACLE I				
VAST BENTTE	A	Big Bin	O finde Tard	wmoone te		1.00		
	MASAS	O loop that BWD-		Automatic In				S MeaniNation
Danio cover	and the second	1BM		C Maler				Duix
Watta Bellen	E cosol			O perceity of	-		B Balagittana hate	States and states
A	Contraction of the second		P	OCIENT	Contraction of the local distance of the loc	10.12.00.000	And the second	the started incom
NOME - MOD		NEWSON DATABASE	-	AL THE TANK			Canada	MATI-
OBAGAS AND	MongoDB	aws - O Ge	mole Cloud	Christense	neodj	ms=	kination	DATAMANTI A ABUTHACTICAS
IBM nel BT	Compe Chul	E Commander O		no Chaine	di Acampiliti		H	ID houses 14 make
awst awst	ORACLE	America es	and the second se	00000 0	AA HIM D	Minut	Chemical B	Crimine M Bauna
	ormer Biblerktogie	CT manners .		A Handata	CHACLE	0	Sale.	prieta
and the second se	None DATASTAN			and and a fight strain		plans M	ATABABLE	
and the second s	an mick and the	4 supabase	and the second sec		Casered 9	4	Process & Or	
	STATE TOURS	Timescole	ProCAP 1	Santa June	APPLIE D		ziliz ißde	ent 🖉 Velpa
	· Quantitati Bar	A Hank 1961	and a second	duidos KX	© Orall		and the second second	
	Table	HEVERHEETL	GATA INTEGAN	TION		DATA DOM	ERRORACE	
1 m 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	end alteryx	Census	SAD Can	a Services 🕀	segment			Collera
Z former ES			· ·····	Boom (1) Harts	· employe			n 😳 data workt
	Arbuta CD Datorios	." hightouch	Provin \$1	EALLAN OIR4	A + Atterbit	1 200	bi metaphor	C hyper lands
	Contraction (Company)	# rudderstack	celly: 0-	until () second	en 13 thattie			Ma OKERA
	Story Statemen	#MessageGours	T Freshpo	sine independent	O unarray			Statt E Seconda
	C Panalas & Prophers	AFOUTOMIC	denodo <sup>®</sup> (	RUX Hootel	ind @ Segar	.153		estor (- Camyai
	Blease ornos	Ontolla	enchy ner	ddata meers	a ar make			Allocare Orige
	0	Qctolis			NLA Y' ROCOR	* roite	Statione Q	OCTOPN
ORDETTATION -	DATAQUALITYA -	FLALT MANAGER	MONT	лионетовека -	- 2	NACY	COMPUTE -	
ASTRONOMER		eldona di missari	Bata FLOA	TADOC BWST	0		ans	aws
Passing Courses	to Coblers u	rovel Man	7. C.	when the Avenue		tent Content	-	and and
Operationals UNION	Same Gas		spin spin spin spin spin spin spin spin	D GOVERNO D		Parman shifting	Real Property lies	22 aws-
No. R. C.	R metaplane (7	siller Ry No	who solarie	ends Cases	-	In St. frantes	PENGUIN	CINIDIA .
Assess Alsess		China Standard			protototo (D)	AND A SHORE	- Martinetter	- suprimer
Google Cloud	talend some (			More More	100 m l 🗰	the granter	Entering Parket	Anirs Moreal
World Date.	munner Gal		N De	AaSet Science		Care Parent	nuclie	0.08
Gargera Miloccu			Vee Vee	MAN Pager	Duty 1	- Arees C	1 INFUTU	a the success

denne Gen Obelernetes die	A Annual of Control A Annual Control A control from Control A control from Control Control from Control from Control Control from Control from Control Control from Control from Control Control from Control from Control from Control Control from Control from Control from Control Control from Control from	Contractions and a strain of the second strain of t	Bane Bane Bane Bane Bane Bane Bane
DATA MARCETPLACES	C. 104 <sup>1</sup> 2 <sup>1</sup>		and the second second

 Matt Turck (@mattturck) , Aman Kabeer (@AmanKabeer11) & FirstMark ( Version 1.0 - March 2024

UN		
Kafka	AUTO SCHENIA	MART
Looker Airtlow	Alt Fivetran	ELINE CONTRACTOR
AUSTER	DATA EXPECTATIONS ark GREAT	
TH DA NO	TA STACK:	TODEN,
	DATA Now W FL	STACKS ITH MORE AME

I have been the second	Constanting of the second	Control Contro	Control aver Annual (2010) Advance foreith appendent (2010) Control (2010) Contro	0 Cabo @ hpuns 1		Stage and
	• renotive at Vendu	CATIONS - H	na feet	annak Titruewi	nd WARNER Class	ing photos

		Annual Annual States	0	The second of th	4		
	ACCINCASE A man call termine denote A man call termine denote A man call termine denote A man call termine denote a man a man call termine denote a ma	APPLICATIONS		ADSCULTURE D		ACROSPACE DETENSION ADVI- CELTENSION ADVI- CELTENSION ADVI- COMPANY ADVI- NOV ADVI- NO	
1441. () 144. () 144. ()		Come Party Come	POINT One	S and a		CHOIS- INCUSTRY QPalaret Elesa	

Constant and a second and a second and a second	TAAII Woodbaard	A bany A	Santalitäisen siinen si	Ø Apache Deppeter
---	-----------------	----------	--	-------------------

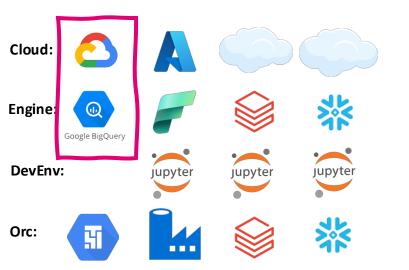
#### DATA & AI CONSULTING

BM	e Consulting	C	in Lee	nsayHer	tz skale	om	٥	泉調	A CO.	TERNA	av 2		- 1.21	G Lighthouse
alabe .	Se TIER	DEVE N	Azoti	adde	opto 6	-	THE	B	ytecod	e IO	\$p	EITDATA	900	E Orange
2-03	T Hann	Ration	() mart	North 1	2 Parfatas	2013	EY	\$5 ·	CapTech	-; A	inpoint .	D4542	A	мттоата

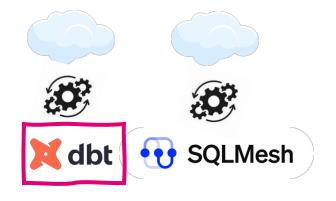
tmarkcap.com

FIRSTMARK

#### execution engine



## sql transformation framework



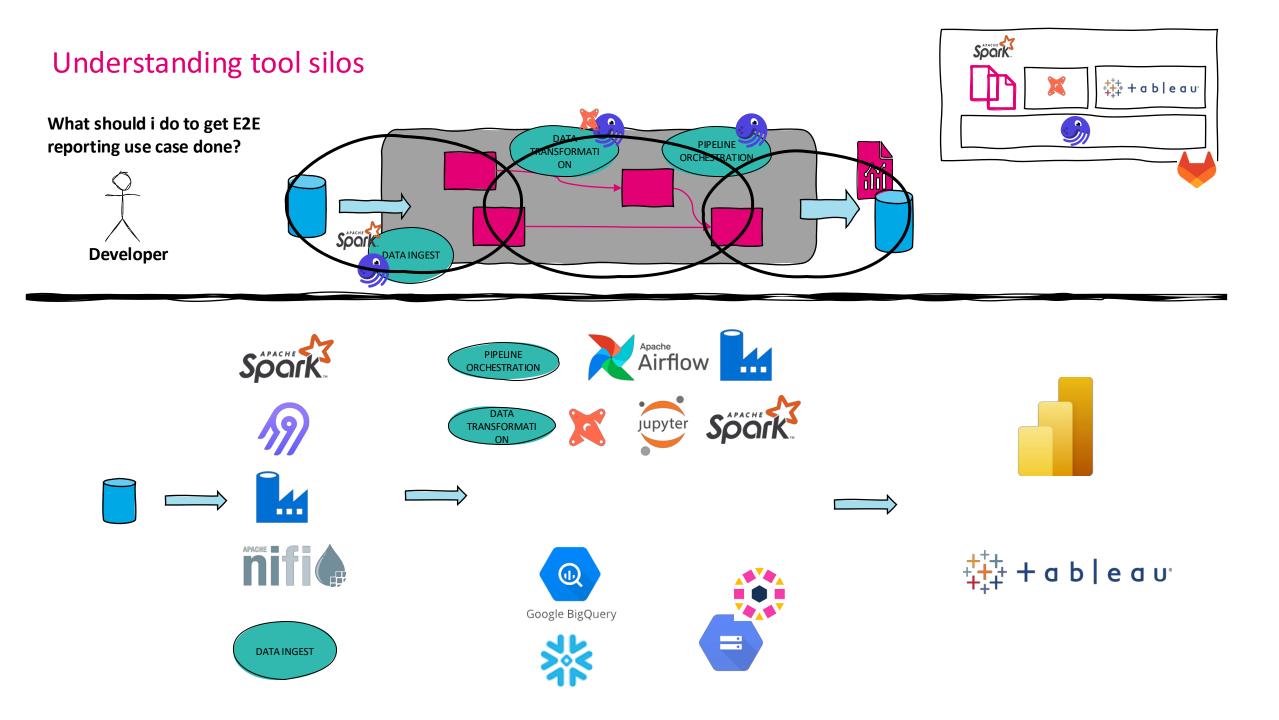
#### orchestration engine



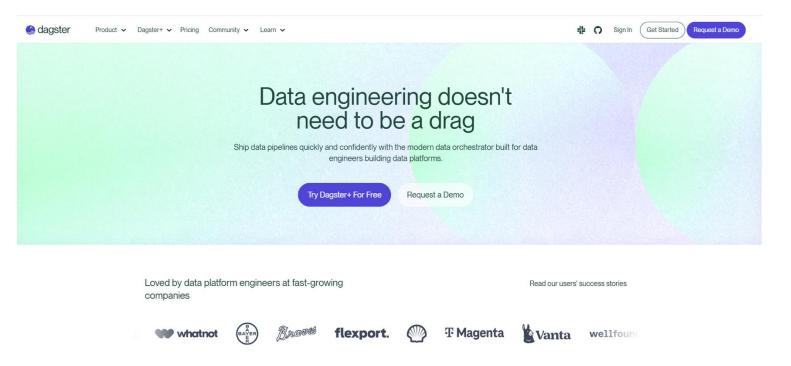
- Full stack offered by one vendor
- E2E integration lock in
- Deployment is always a workaround
- No SWE, no local development
- Orchestrator is second class citizen and always task based
- + Frontier in the lakehouse approach
- + Notebooks environment is very convenient
- + Everything on one place

- Orchestration is just for DWH/SQL part of the platform
- + Frontier in the SQL development+ SWE for DWH development

- More technical knowledge needed to setup and use correctly
- + Integration is important
- + full SWE and local development



#### Dagster as the core of the platform



- at Magenta we decided to build around the orchestrator and not around a execution engine
- hybrid deployment controlplane SaaS runtime in our k8s
- software engineering best practices for project development and deployment
- asset-based mindset for data flows (graph like a calculator for data dependencies)
- new concepts in orchestration

## New enabled concepts

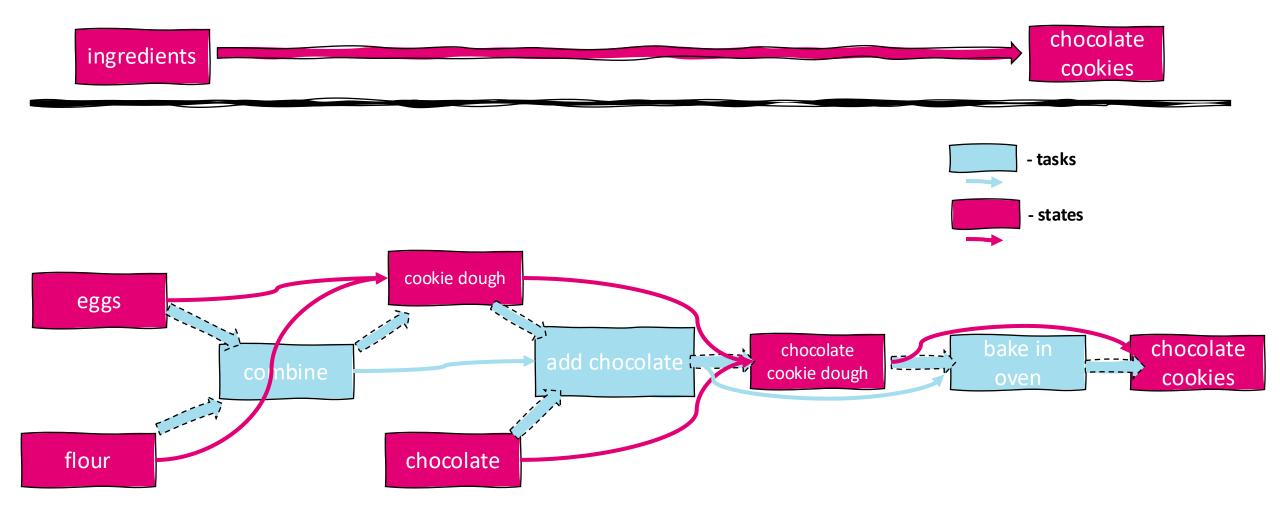
Asset based graph

. . . .

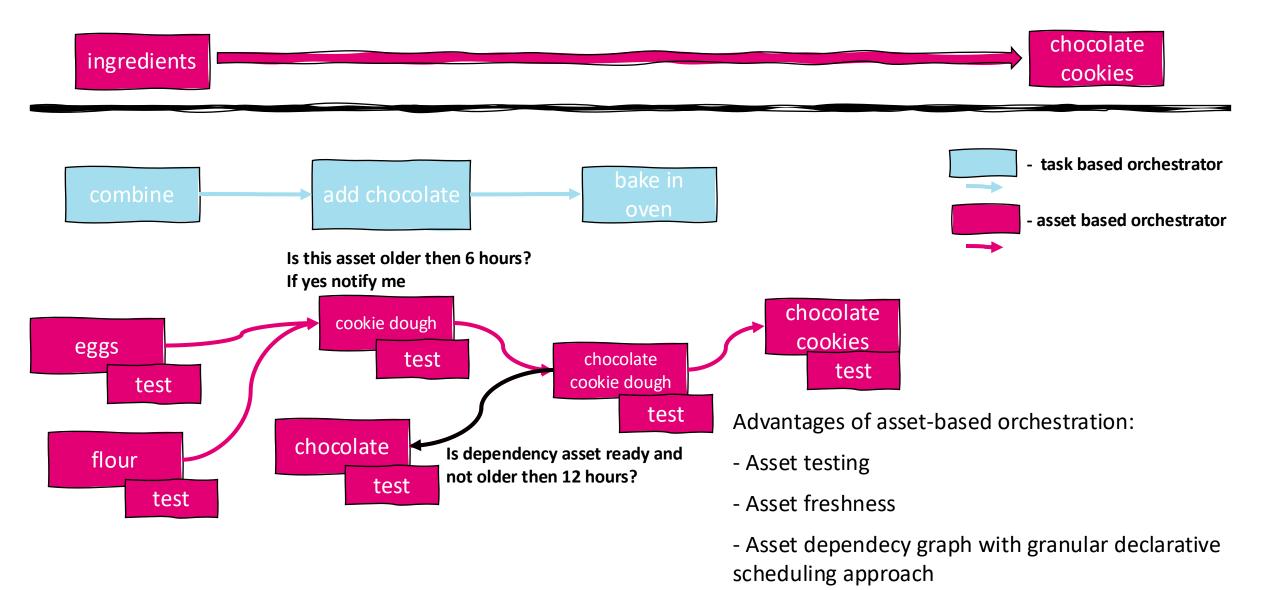
- Metadata driven pipeline creation
- Reusable Components

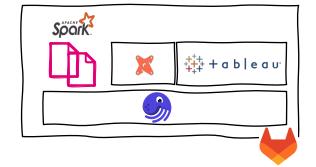
Catalog > All assets > my_f	irst_extra	ct_dataset 🛛 🦻 New in branch	External Asset				
verview Events Checks Lineage	Insights						
↔ Nearest Neighbors ← Upstream	I→ Downstrea	am Graph depth - 4 + All	Column Select a column				+‡ Materialize all (5
jdbc_to_bigquery				data_product_example	default dp-sample		
JP New in branch		å <sup>₽</sup> New in branch	5º New in branch	الله الله الله الله الله الله الله الله	3 <sup>9</sup> New in branch	3 <sup>9</sup> New in branch	ارم New in bran
E my_first_ingest_source	B	∃sample_ingest_asset	<pre>maintingest_asset_staged</pre>	mw_first_dbt_model_v2 A starter dbt model	my_first_extract_dataset	🖼 my_first_sheet	my_first_dashboard
ddwh2_taf.dbtpoc.my_first_ingest_source	1	Asterialized 26. Feb., 08:36	dbt model sample_ingest_asset_staged , Never materialized	Never materialized .	No description	No description Never materialized	No description Never materialized
l-			M dbt 🙆 BigQuery			Tableau	di Table

#### Asset and Task based orchestration: Chocolate cookie example

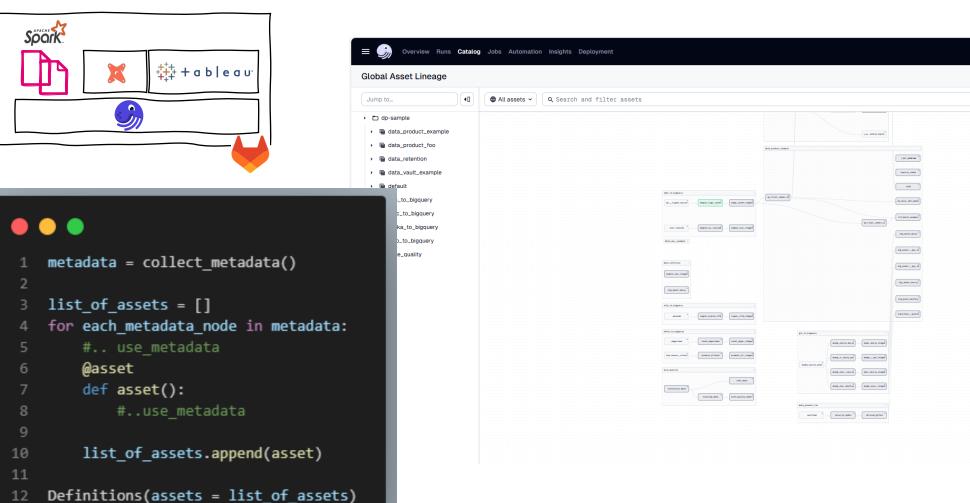


#### Asset based orchestration





= 🧊 Overview Runs Catal	g Jobs Automation Insights Deployment		٩	이 #90 🖄 11 feat/add-table > ⑦ 🔺
Global Asset Lineage				Reload definitions
Jump to		ts		↔ Materialize all (77) •
<ul> <li>L dp-sample</li> <li>G data_product_example</li> <li>G data_product_foo</li> <li>G data_retention</li> <li>G data_vault_example</li> <li>G default</li> <li>G cs_to_bigquery</li> <li>G kafka_to_bigquery</li> <li>G sfto_to_bigquery</li> <li>G wine_quality</li> </ul>			I ANNA ANY ANNA ANY ANNA ANY ANNA ANY ANNA ANY ANNA ANY ANNA ANY ANNA ANY ANNA ANY ANNA ANY	



Q

Ō

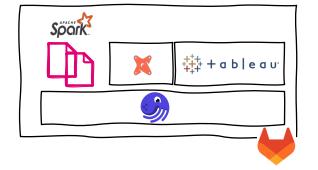
Q 8

۲

Q 🔘 #90 🖄 feat/add-table... 🗸 ⑦

😪 Reload definitions

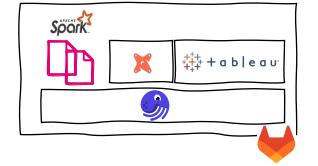
↔ Materialize all (77)...



#### •••

```
1 metadata = collect_metadata()
2
3 list_of_assets = []
4 for each_metadata_node in metadata:
5  #.. use_metadata
6  @asset
7  def asset():
8      #..use_metadata
9
10 list_of_assets.append(asset)
11
12 Definitions(assets = list_of_assets)
```

1	<pre>configuration_files = read_ingest_configuration_folder(path)</pre>	Q O #90 2 12 feat/add-table > O A
2	list_of_assets = []	G 👫 Materialize all (77) 🔹
3	for each_config_file in configuration_files:	
4	<pre>config = parse_config(each_config_file)</pre>	
5		
б	@asset(	
7	name = config.name	
8		
9	<pre>def ingest_asset():     df</pre>	
10 11	<pre>df = spark.read(config.source) df.write(config.source)</pre>	
12	list_of_assets.append(ingest_asset)	
13	TTACTOL TOTAL CONTRACT (TUBER CTORE)	
14	Definitions(assets = list_of_assets)	
		<b>e</b> ,
	There are an	

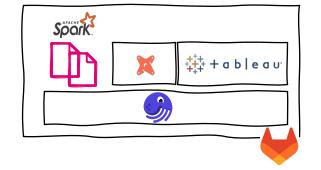


#### •••

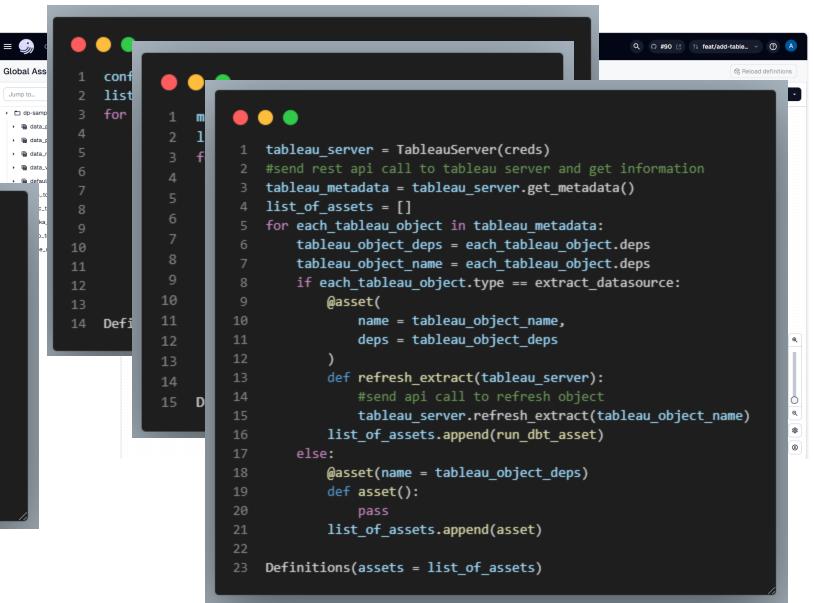
1	metadata = collect_metadata()
2	
3	list_of_assets = []
4	<pre>for each_metadata_node in metadata:</pre>
5	# use_metadata
6	@asset
7	def asset():
8	#use_metadata
9	
10	list_of_assets.append(asset)
11	
12	<pre>Definitions(assets = list_of_assets)</pre>

	• •		Q 🔿 #90 🖻 🕅 feat/add-table > 🕜 🧖
1	conf		Reload definitions
2	list		✓ ★ Materialize all (77)
3	for	1 manifest = read_dbt_manifest(path)	
4		<pre>2 list_of_assets = []</pre>	
5		<pre>3 for each_node in manifest:</pre>	
		<pre>4 model_name = each_node.name</pre>	
		<pre>5 model_deps = each_node.deps</pre>	
		6 @asset(	
		<pre>7 name = model_name,</pre>	
		<pre>8 deps = model_deps</pre>	
		9 )	
13		<pre>10 def run_dbt_asset(dbt: DbtClient)</pre>	
14	Defi	<pre>11 dbt.run(f"select {model_name}")</pre>	
		12	
		<pre>13 list_of_assets.append(run_dbt_asset)</pre>	
		14	
		<pre>15 Definitions(assets = list_of_assets)</pre>	
	3 4 5 7 8 9 10 11 12 13	3 for 4 5 6 7 8 9 10 11 12 13	<pre>1 conf 2 list 3 for 1 manifest = read_dbt_manifest(path) 4 2 list_of_assets = [] 3 for each_node in manifest: 4 model_name = each_node.name 5 model_deps = each_node.deps 6 @asset( 7 name = model_name, 8 deps = model_deps 9 ) 13 def run_dbt_asset(dbt: DbtClient) 14 Defi 11 dbt.run(f"select {model_name}") 12 13 list_of_assets.append(run_dbt_asset) 14</pre>

= 🌍

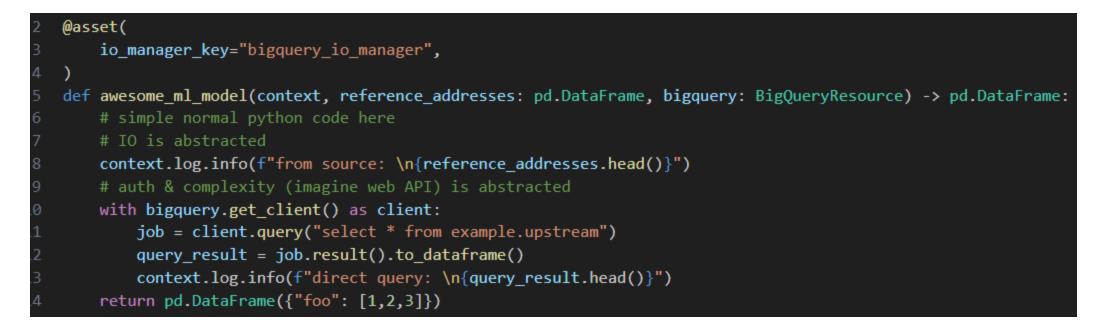


1	metadata = collect_metadata()
2	
3	list_of_assets = []
4	<pre>for each_metadata_node in metadata:</pre>
5	# use_metadata
6	@asset
7	def asset():
8	#use_metadata
9	
10	list_of_assets.append(asset)
11	
12	<pre>Definitions(assets = list_of_assets)</pre>



#### **Reusable components**

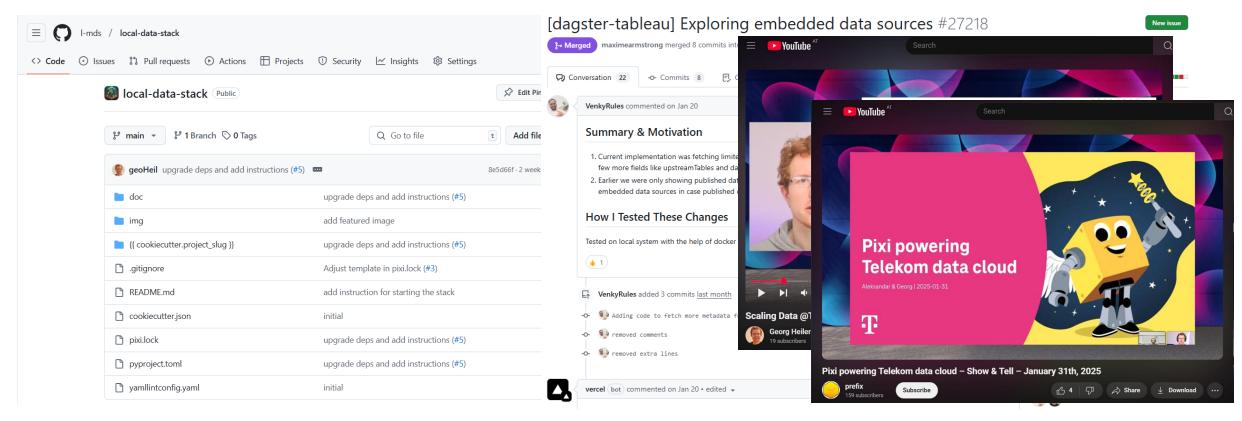
- Define once, test & reuse
- Resources  $\rightarrow$  Encapsulate complex logic to interact with external systems
- IO manager  $\rightarrow$  Make complex IO interactions substitutable & testable
- Benefits
- Dependency injection
- Day 1 productivity: Scale the data pipeline down to a single laptop
- Increase self-service: Business/DS focus not required to handle complex IO



#### Takeaways

- Integrated asset-based graph is key (from ingest, transformation, reporting, tests to AI)
- Event driven connection
- Better collaboration (scaling)
- Software engineering principles enable business self service
  - Blueprint
  - Automate all the things: CI/CD (stateful & stateless)
  - DRY: build tested foundation dependency injection
  - Make business departments part of the key processes and pipelines
- Executable specification (metadata, contracts)
  - Interface Mangement
  - Preserve semantics
  - Preserve compliance (security classification, PII, retention)

#### Last things



## Data platform is team work and Pixi powering Telekom data cloud we are very proud and excited about the jurney ahead

Scaling data pipelines @Telekom

# Scaling data pipelines @Telekom

geoheil.com/event/magenta-data-architecture-25

