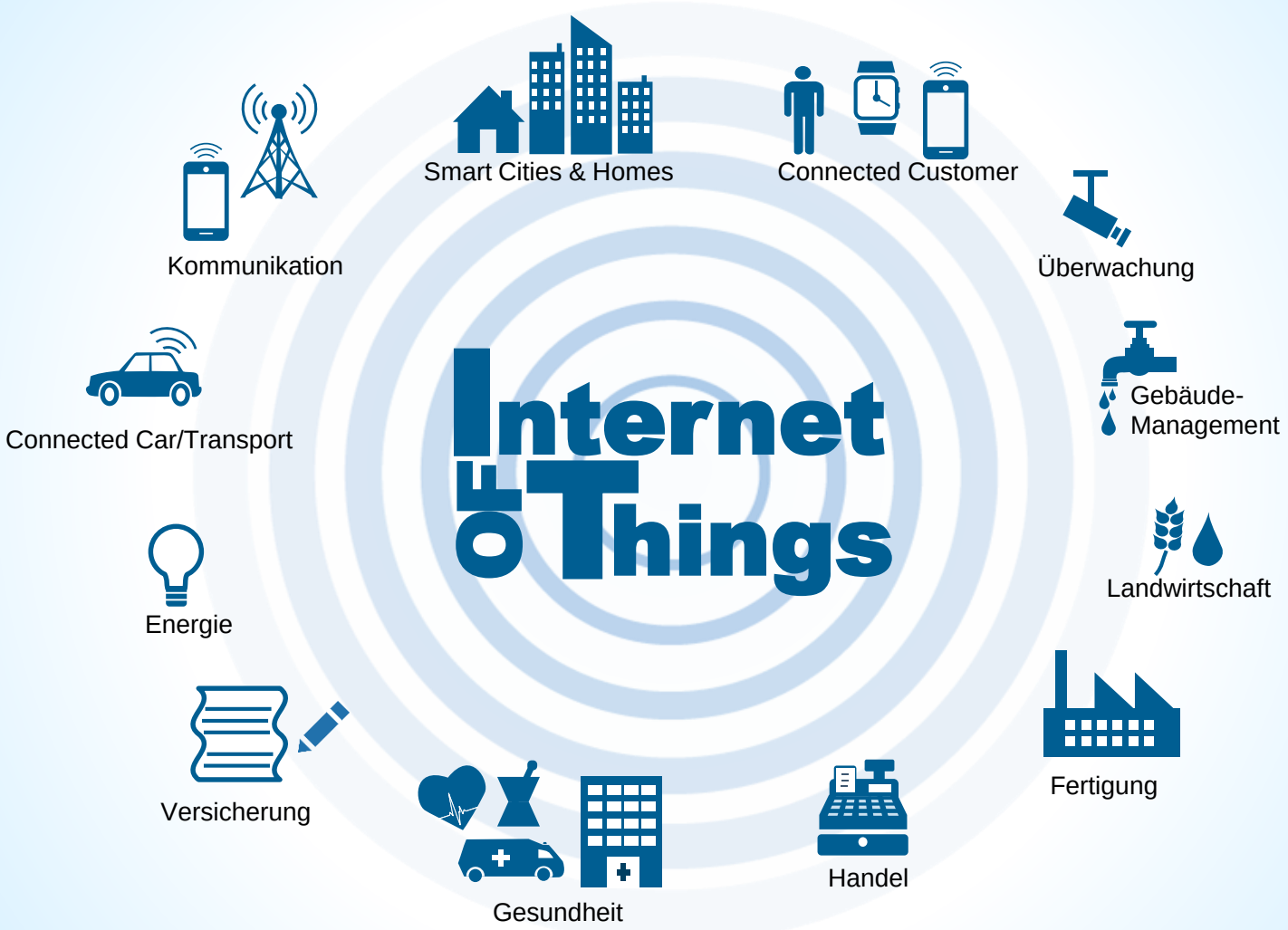
SAS EVENT STREAM PROCESSING - BEST PRACTICES IN DER ANBINDUNG UND ANALYSE VON ECHTZEITDATEN

PHILLIP MANSCHEK, SAS SYSTEMARCHITEKTUR

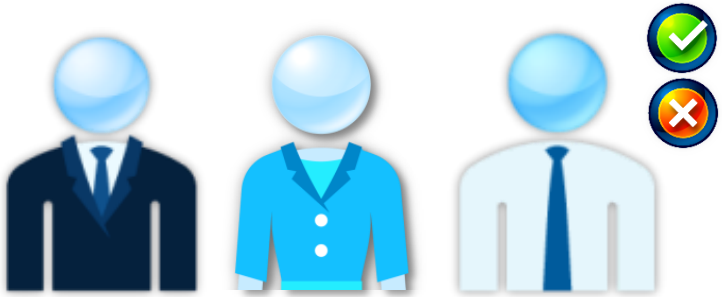
AGENDA

- Warum Streaming Analytics?
- SAS Event Stream Processing
- Demo
- Best Practices



IOT + BIG DATA | NEUE ANFORDERUNGEN

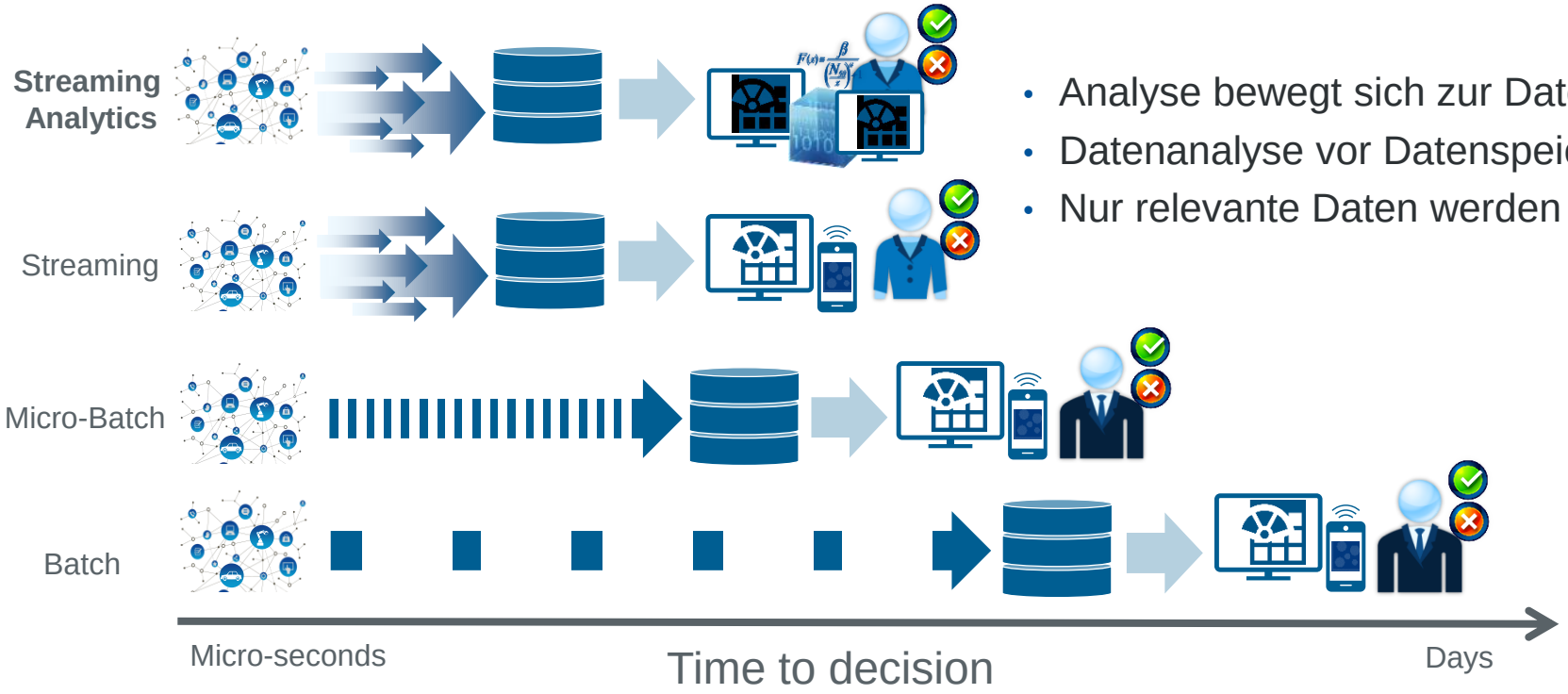
Volume
Velocity
Variety



- Unmittelbare Antworten (Echtzeit).
- Minimierung der Zeit zur Entscheidungsfindung.
- Kontinuierlich Risiken und Möglichkeiten prüfen.
- Agile und leicht zugängliche Prozesse.
- Big Data “V”s sind Grundlage, keine Besonderheit.

NEUE ART DER INFORMATIONS-VERARBEITUNG

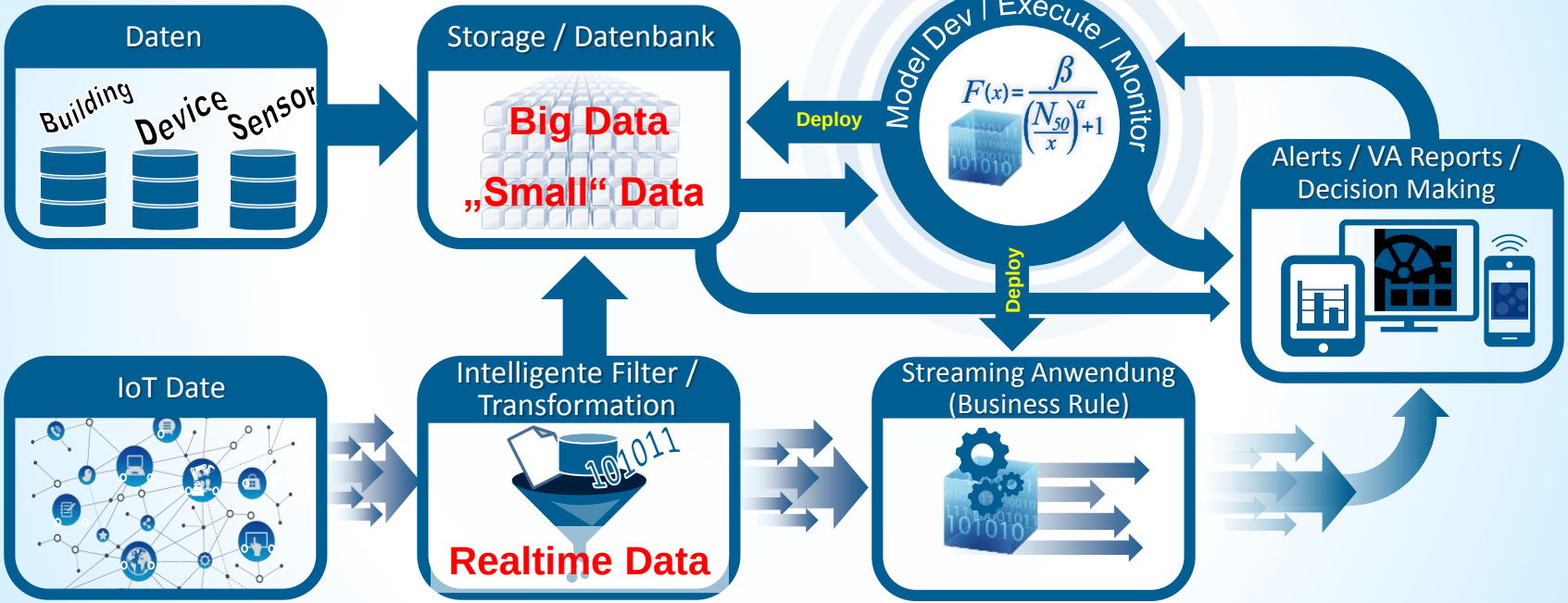
ANWENDER BENÖTIGEN UNMITTELBARE ANTWORTEN

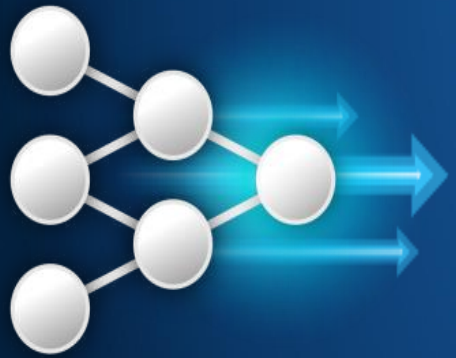


- Analyse bewegt sich zur Datenquelle.
- Datenanalyse vor Datenspeicherung.
- Nur relevante Daten werden behalten.

Verarbeitung von Streaming-Daten bedeutet **unmittelbare Antworten** und damit **schnellere Entscheidungen.**

ARCHITEKTUR | EIN ANALYTISCHES MODELL - DREI ANWENDUNGSFÄLLE

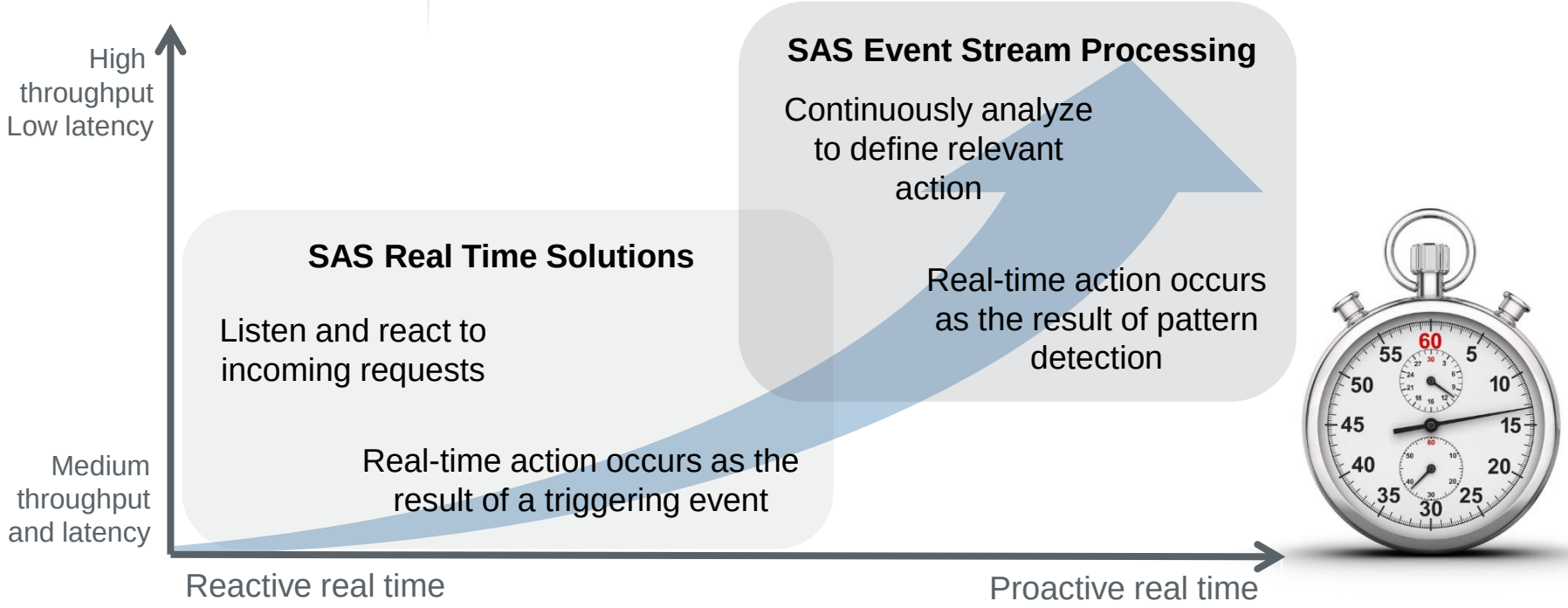




SAS Event Stream Processing

SAS® EVENT STREAM PROCESSING

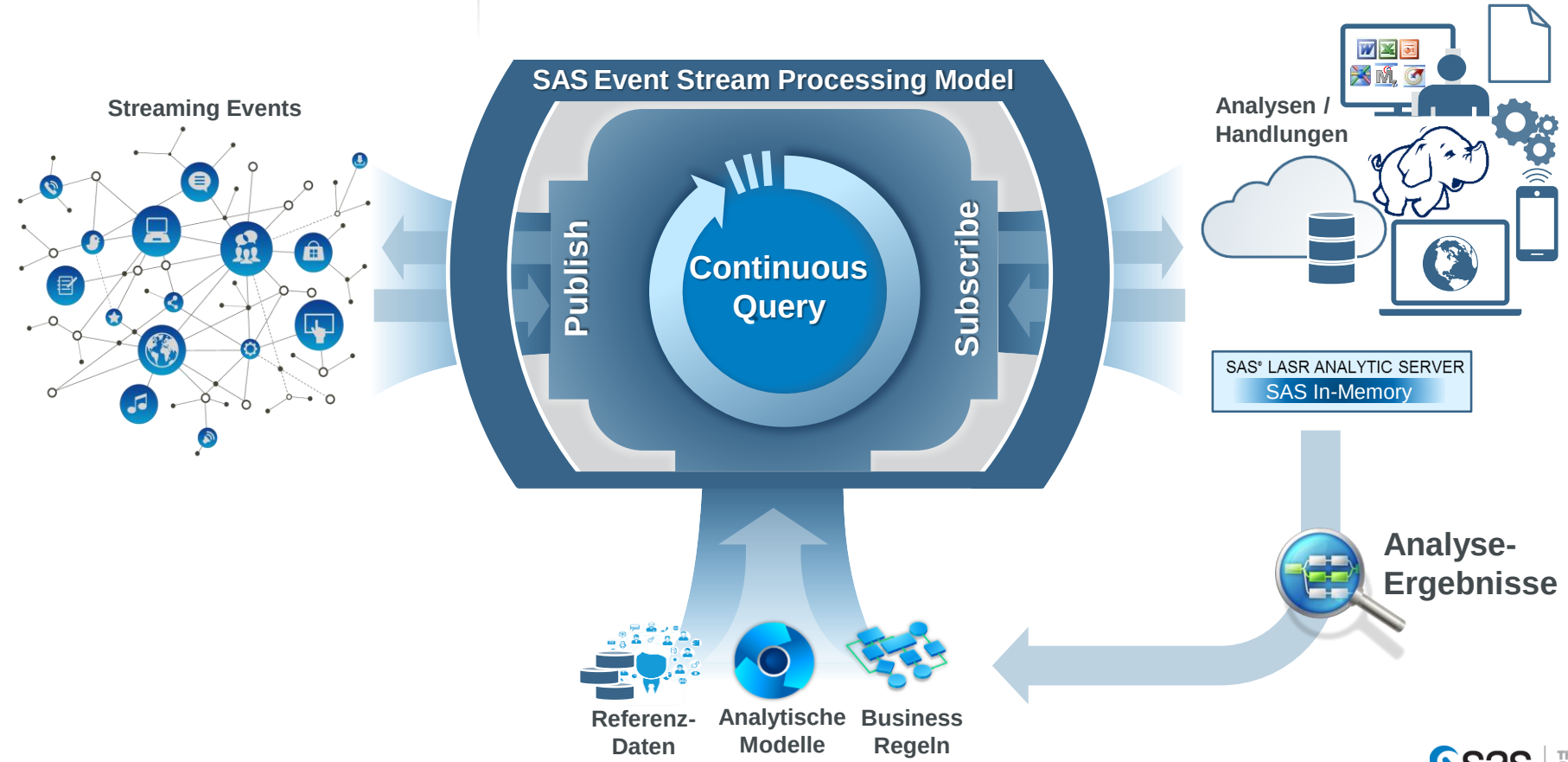
PROVIDES REAL-TIME ANSWERS



Processing high throughput, low latency streaming events requires to move from reactive real time to proactive real time

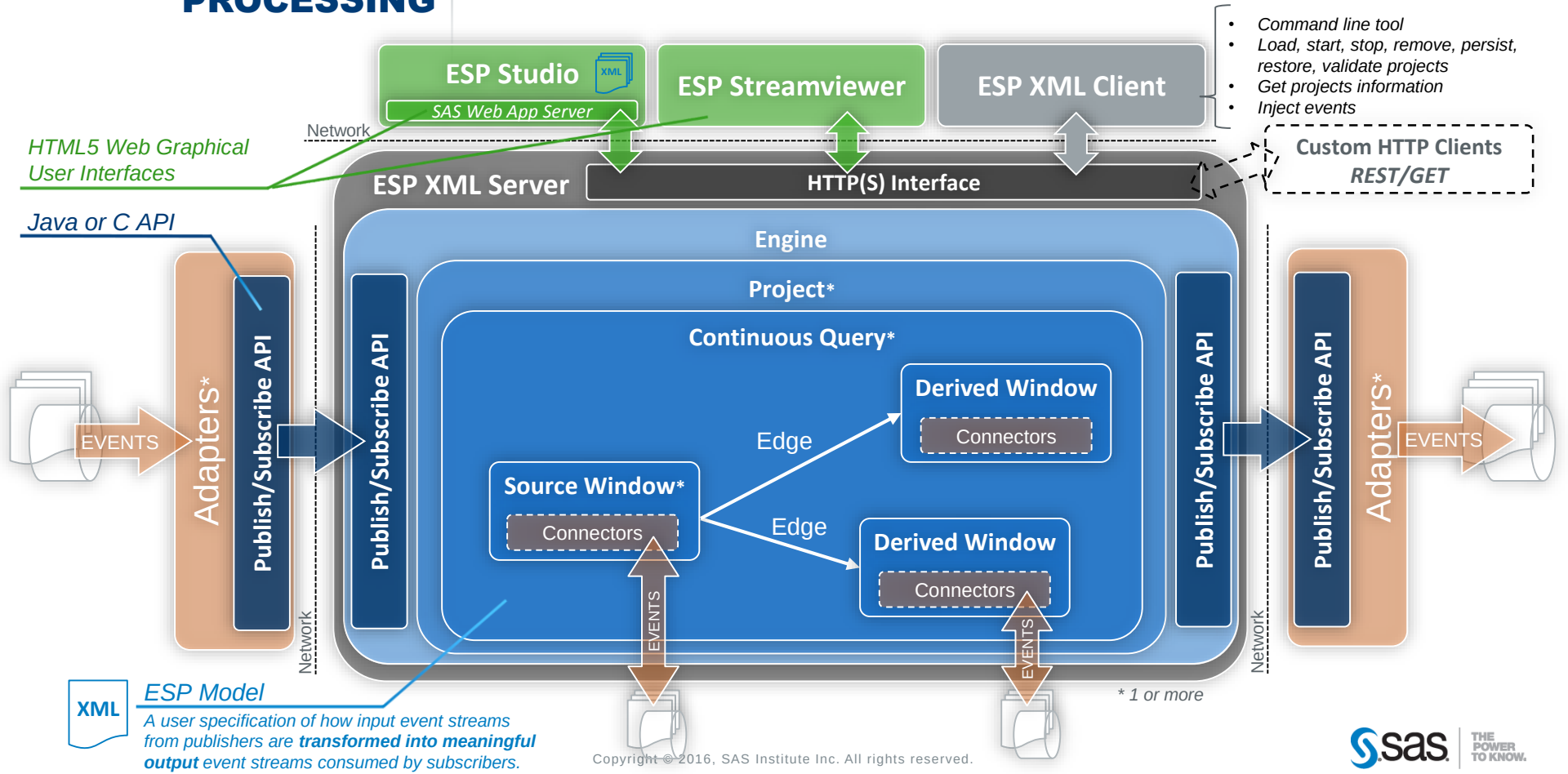
SAS EVENT STREAM PROCESSING

DER STREAMING-LEBENSZYKLUS



sas THE POWER TO KNOW.

SAS EVENT STREAM PROCESSING



SAS EVENT STREAM PROCESSING

ENGINEERED FOR SPEED, AGILITY & ANALYTICS

SAS Event Stream Processing bietet:

- **Millionen Events pro Sekunde** Durchsatz.
- **Millisekunden** Antwortzeit.
- Auf Standard **Commodity Hardware.**

Continuous in-memory processing

OS native application (Windows, Linux)

Guaranteed Delivery

Linear scalability

Fastest ESP on the market

Lightweight embedding technology

Cloud ready (Cloud Foundry with BOSH or Chef)

Clustering / Failover

Dynamic model update

SAS Data Quality

SAS Text Analytics

NLP methods

K-Means, DBSCAN

SAS DataStep and DS2

Open source integration

SAS business solutions integration

Dataflow centric modeling

Drag & drop visual modeler

Visual, XML or C modeling

Publish & Subscribe API (Java, C, Python)

SAS EVENT STREAM PROCESSING

INTEGRATION (300+ ENDPUNKTE)

SYSTEMS & APPLICATIONS

Solace Systems
TERADATA
GREENPLUM DATABASE
TERVELA Big Data in Motion
boardreader*
WebSphere MQ
TIBCO RendezVous
PI SYSTEM OSIsoft.
Axēda
twitter
spryware
ORACLE

OPEN SOURCE

RabbitMQ
nifi
hadoop
protobuf Protocol Buffers
kafka
Apache Camel
python
MAPR
Hortonworks
CLOUD FOUNDRY
BOSH The outer shell for cloud software
Chef
cassandra*
YARN READY
FLUME*

STANDARDS

FILE/SOCKET	JMS	HTTP RESTFUL
XML / JSON	MQTT	SMTP
ODBC	SYSLOG	NETWORK SNIFFERS
	DB LOG SNIFFERS	WEB SERVICES

PUBLISH & SUBSCRIBE API

CONNECT TO ANY SYSTEM WITH JAVA, C, PYTHON
FULLY DOCUMENTED AND EASY TO USE

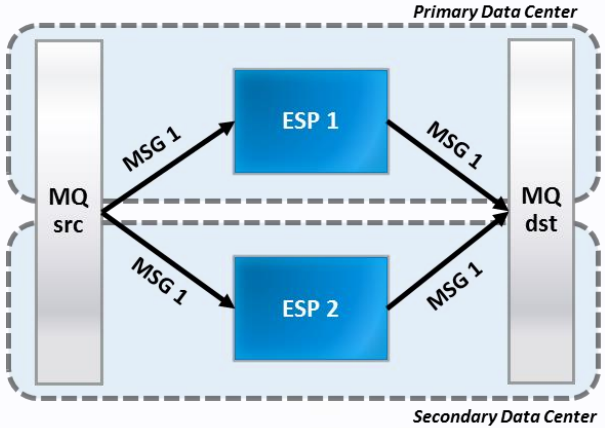
DEMO

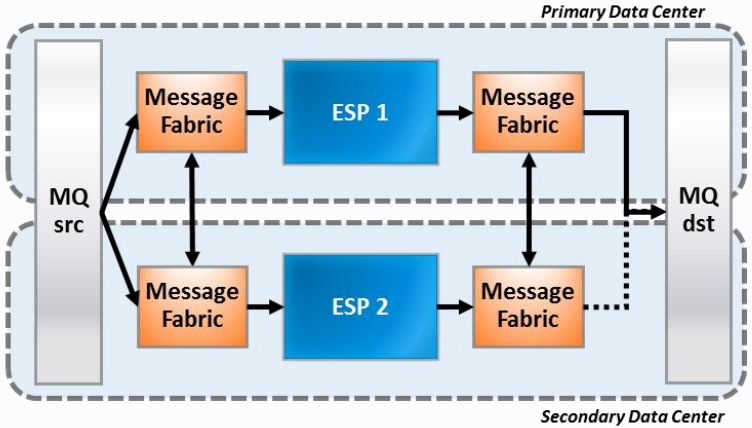
- https://www.youtube.com/watch?v=3xW-hUDsRIQ

BEST PRACTICES

GÄNGIGE FALLEN, WICHTIGE ÜBERLEGUNGEN

Streaming-Anwendungsfälle sind oftmals ‚mission-critical' ➔ Hochverfügbarkeit





Hochverfügbarkeit:
- N+1 Failover
- Exactly once vs. at-least once
- State persistence
- Message queue mit Buffer

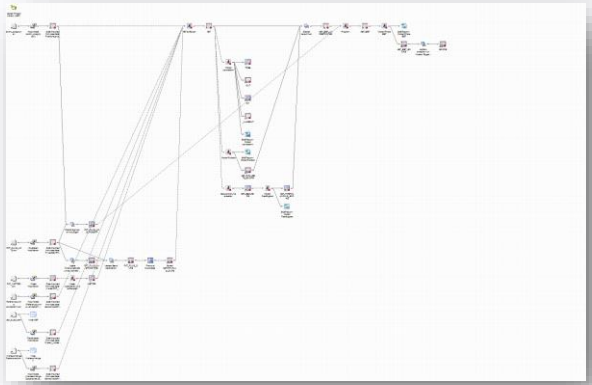
Robustheit:
- Engine hält Daten im RAM
- Swapping führt zu Performance-Einbußen; Engine bleibt ‚up'
- ESP-Modelle können zur Laufzeit verändert werden

Performance:
- Lineare Skalierung möglich
- Core-Affinität (NUMA) wichtig
- Adaptoren können auf andere Maschinen ausgelagert werden
- Hadoop als Datenspeicher ideal

BEST PRACTICES

IMPLEMENTIERUNG - VORGEHENSMODELL

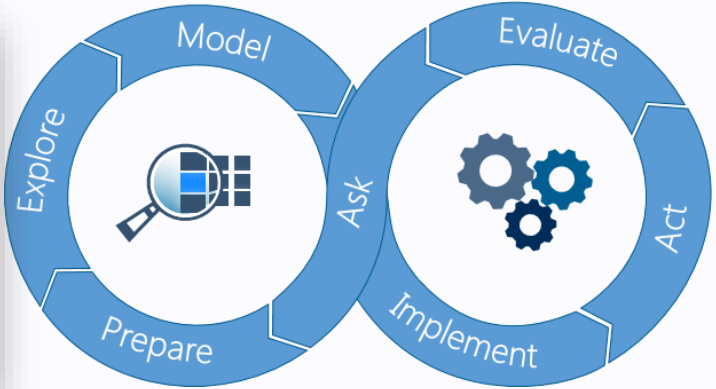
Modell Prototyping in SAS EG / SAS EM



Batch ist anders als Streaming

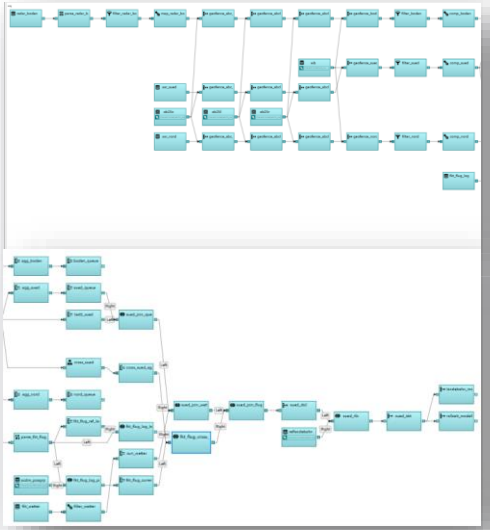
- Daten sind begrenzt
- Zeitlich nicht geordnet
- Historie ist vorhanden
- Alle Daten im Zugriff
- Keine laufzeitbedingten Einschränkungen

Entwicklung



Operationalisierung

Implementierung in ESP



- Komplexe Modelle verschachteln
- Wo möglich auf ESP Funktionen setzen (z.B.: XML Parsing)
- Datenaufbereitung nach Möglichkeit komplett in ESP

SAS EVENT STREAM PROCESSING

SKALIERBARKEIT

Beispiel: 32 Core Linux-Server

Event Rate (events per second)	Event Feeding Elapsed Time (MM:SS)*	Elapsed Time of IoT Adapters (MM:SS)		
		File and Socket Adapter	LASR Adapter	HDFS Adapter
100,000	16:06	16:06	16:06	16:06
300,000	05:22	05:22	05:22	05:22
500,000	03:13	05:50	04:55	03:13
700,000	02:18	05:50	08:40	02:18
800,000	02:00	05:50	10:10	02:00

	Instances	Event Rate (x 1,000 events per second)	Average CPU Percentage
K-means	1	738	59
	2	1,457	61
	4	2,768	58
	8	4,588	40
Broker Surveillance	1	527	50
	2	1,000	50
	4	1,930	43
	7	3,182	46

MONITORING

Herausforderung: Sicherstellung der korrekten Funktionsweise (technisch und fachlich)

Technisch:

Allokation von Ressourcen:	CGROUPS, ulimit, YARN
Monitoring:	OS Boardmittel, Logfiles, Streamviewer, BOSH (Cloud Foundry), „indirekt“ am Empfänger
Deployment:	Chef, BOSH, XML Files

Fachlich:

Anbindung an SAS LASR Server (VA als Dashboard)
Anbindung an Langzeitspeicher (Flatfiles, Hadoop, …)

ERWEITERBARKEIT, AUSBLICK

Erweiterbarkeit mittels Procedural Window
- C++
- SAS DS2
- SAS Data Step
- Micro Analytics Service (DS2 oder Python)

Roadmap
- Edge Analytics
- Cloud
- Weitere Datenquellen
- Weitere analytische Verfahren (R?)

ZUSAMMENFASSUNG

… es geht darum **Analytics** auszuführen, während die Daten **in Bewegung** sind.