DATA+AI
SUMMIT EUROPE
FORMERLY SPARK+AI SUMMIT

Building a Cross Cloud Data Protection Engine

Richard Conway
CEO and Founder
@azurecoder

Sandy May
Lead Data Engineer
@spark_spartan

Speaker Bio

Richard Conway - @azurecoder

Microsoft Azure Most Valuable Professional

Microsoft Regional Director

UK Azure User Group Co-Organizer and Co-Founder

Data Science London Co-Organizer

Worldwide technology speaker

Passionate about big data, AI and security in Microsoft Azure

Speaker Bio

Sandy May - @spark_spartan

Databricks Champion

Data Science London Co-Organizer

Tech speaker across the UK

Passionate about Apache Spark, Databricks, AI, Data Security and Reporting platforms in Microsoft Azure

Agenda

Richard Conway

What is a Data Protection Engine and Why do we need it? Let's also look at some architecture

Sandy May

Building a simple Data Protection Engine, from the ground up to cover GDPR and give the business a starting point

Data Protection Engine Overview

@azurecoder @spark_spartan

What is the problem?

- GDPR & CCPA fines can be in billions $ now
  - British Airways €204m July 2019 – 500,000 effected customers
  - Highest theoretical fine $21b based on 4% 2019 revenue
- Off the shelf products are expensive
  - With Slow delivery roadmaps that you can't control
  - You still must pay to run them in cloud = more $$$
- Products don't mitigate risk, you still own risk
  - You are responsible to run products over your data
  - Some products won't even own liability for bugs in their software
  - Most don't "detect" PII within data

Should we Build or Buy?

Build

- Own the IP
- Prioritise the features you want
- Built for your use case
- No licence fees
- Use your core technology

Buy

- May have track record
- Bugs fixed by vendor
- Features not thought about by business
- Service Level Agreements

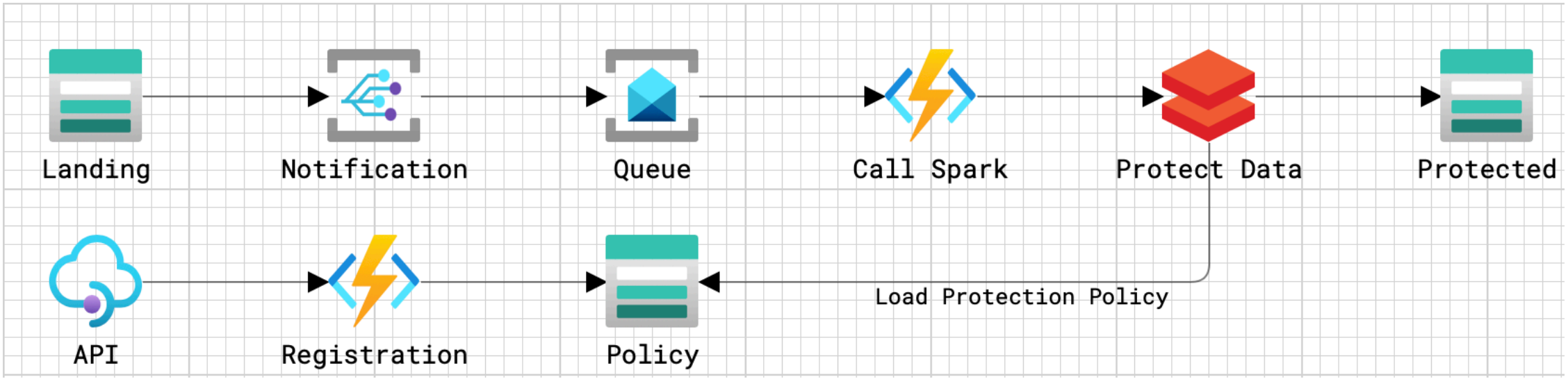
Business Needs

- Run as part of Data Pipeline and ad-hoc
- Track lineage of Data Protected
  - Use a metadata store for all transformations
- Support Pseudonymisation, Anonymisation & Generalization
  - Re-identification required from Pseudonymisation
  - Joining datasets required from Pseudonymisation
- Allow Pseudonymisation Tokens to be migrated to another solution

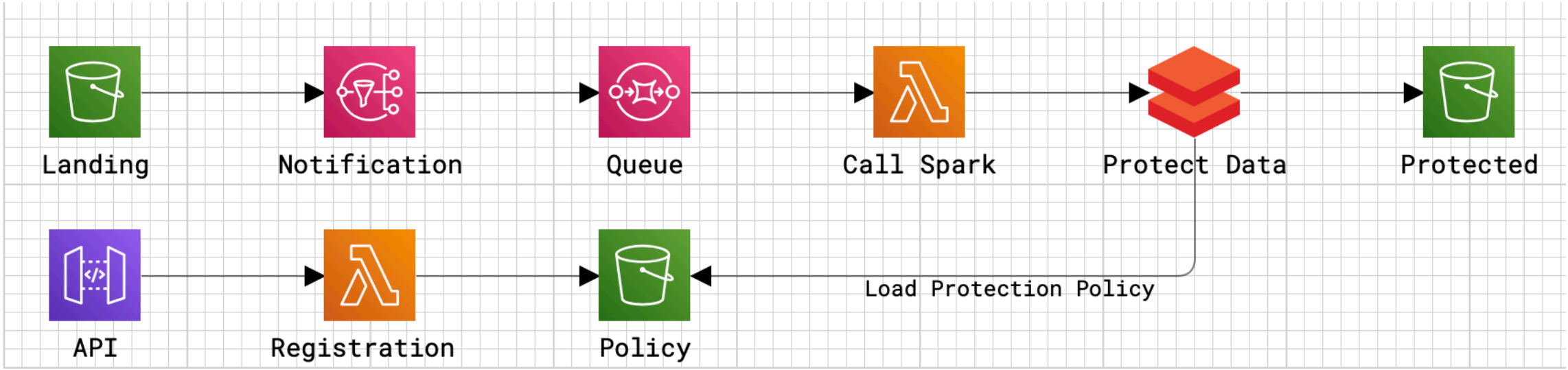
Key Design Decisions

- Support to Run On-Premise and Cloud
- Use Native tools in Azure and AWS
- Token Vault consistency and auditability
- Single Reporting Platform
- Metadata driven

Architecture - Azure



Architecture - AWS



Config Driven Design

c > Users > rusarg > Desktop > DtreePMMLxml

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<PMML xmlns="http://www.dmg.org/PMML-4_2" xmlns:data="http://jpmml.org/jpmml-model/InlineTable" version="4.2">
    <Header copyright="Copyright (c) 2020 ivnard" description="Decision Tree Model">>
        <Application name="JPMML-SparkML" version="1.5.4"/>
        <Timestamp>2020-03-21T13:30:39Z</Timestamp>
    </Header>
    <DataDictionary>
        <DataField name="BAD" optype="categorical" dataType="integer">
            <Value value="0"/>
            <Value value="1"/>
        </DataField>
        <DataField name="MORTDUE" optype="continuous" dataType="double"/>
        <DataField name="YOJ" optype="continuous" dataType="double"/>
        <DataField name="DEROG" optype="continuous" dataType="double"/>
        <DataField name="DELINQ" optype="continuous" dataType="double"/>
        <DataField name="CLAGE" optype="continuous" dataType="double"/>
        <DataField name="NINQ" optype="continuous" dataType="double"/>
        <DataField name="CLNO" optype="continuous" dataType="double"/>
        <DataField name="DEBTINC" optype="continuous" dataType="double"/>
    </DataDictionary>
    <TreeModel functionName="classification" splitCharacteristic="binarySplit">
        <MiningSchema>
            <MiningField name="BAD" usageType="target"/>
            <MiningField name="MORTDUE"/>
            <MiningField name="YOJ"/>
            <MiningField name="DEROG"/>
            <MiningField name="DELINQ"/>
            <MiningField name="CLAGE"/>
```

Let’s Build it!

Summing Up

Future Work

- Machine Learning PII Detection
- K-Anonimity
- Batching Service
  - Databricks cluster only runs one job with 3-5 minute spin up time
  - Delta Lake ensures ACID transaction on requests originating from a single cluster
- De-centralize the solution
  - Allow individual data teams to control their own data protection and pay for their usage
  - Maintain a central reporting solution for business
  - Consideration needs to be given to joins across tokenized data

Conclusions

- Building can be quick and secure
- Prioritise your own business needs
- Can be used as a stop gap while you create a service for an off the shelf product
- No false promises of protection, you control all

Thanks for listening!
Questions?

Feedback

Your feedback is important to us.

Don’t forget to rate and review the sessions.