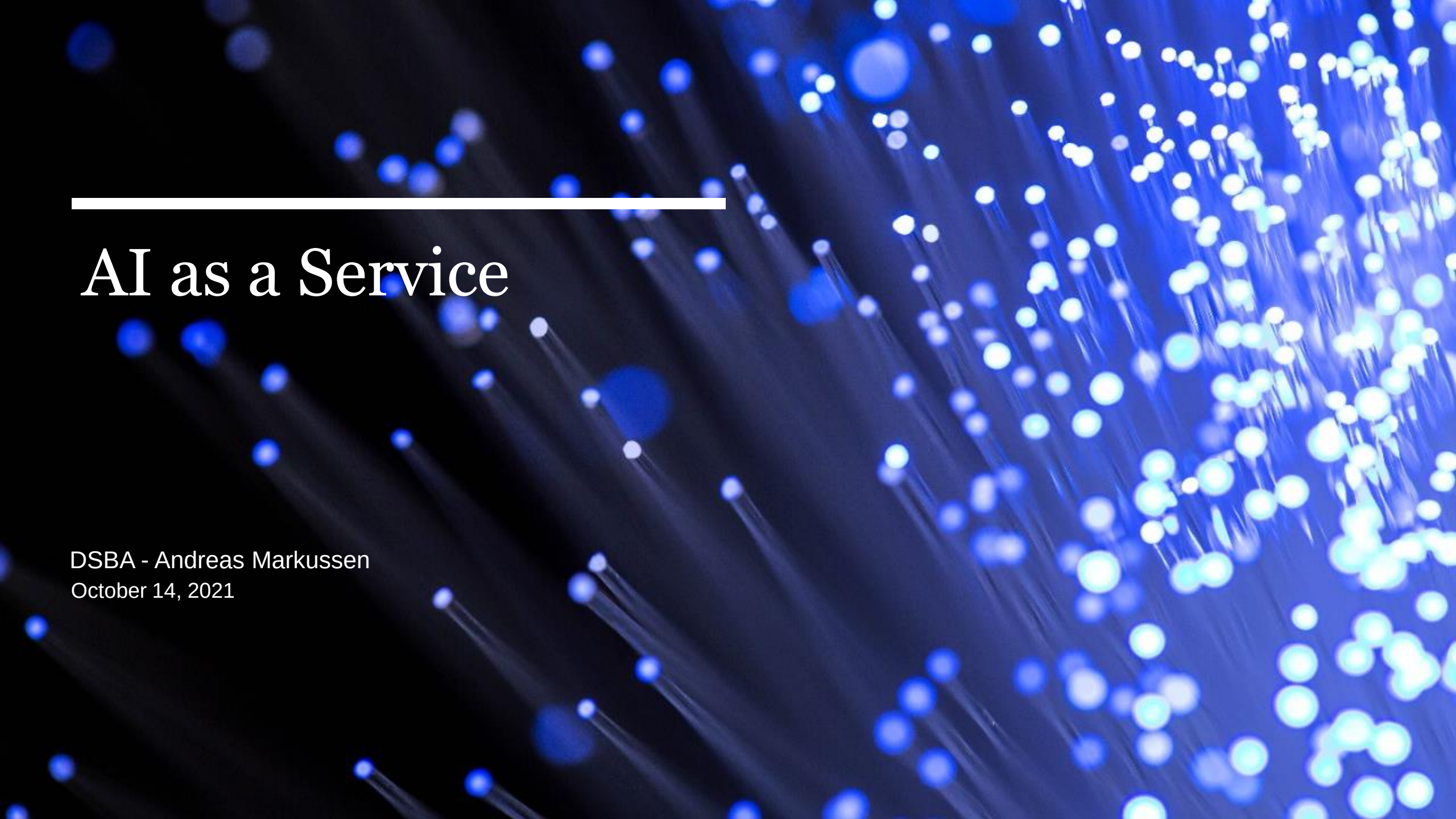


AI as a Service

DSBA - Andreas Markussen

October 14, 2021

Data science can, but should not, stand alone

1. People want Artificial Intelligence, but only if it is convenient
2. AI is a inherently a decision-making tool

Name of the game:

How to we enable non-AI literate people to use our results?

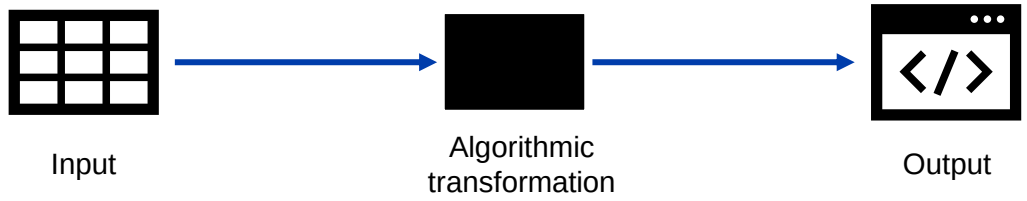
The internet is the ideal delivery platform, although it needs a bit of padding

Relevant use cases for internet as a delivery platform for AI:

- Commercial (internal and external)
- Academic
- Personal
- Patchwork in systems integration

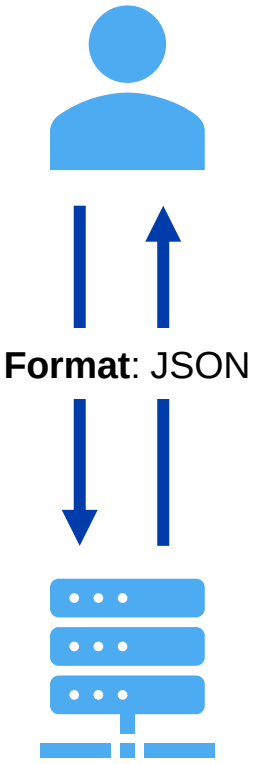
Static models ensure stable content delivery across devices, platforms and locations
– no more notebooks, required IT systems and *”It works on my computer”*

Both the internet and AI takes input, conducts black box transformation and outputs a result:



Information exchange can easily be defined...

APIs return the data you ask for



You send:

URL

Metadata header

(Optional) Authentication

(Optional) Arguments

You recieve:

Response code

Metadata header

Data

The HTTP protocol and making *requests*

GET

Get data without altering it

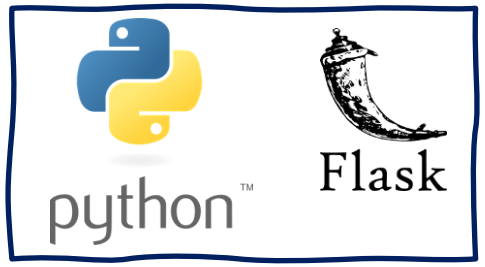
POST

Modify or add data

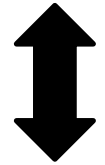
… although there are many elements to a web application

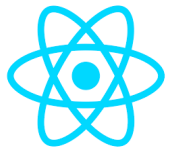
Serverside
Server configuration

Backend
Logical operations



Frontend
Visuals





*This is an oversimplified overview. Most often, a database is also implemented

Luckily, we can disregard or outsource many things

Serverside
Server configuration

Backend
Logical operations

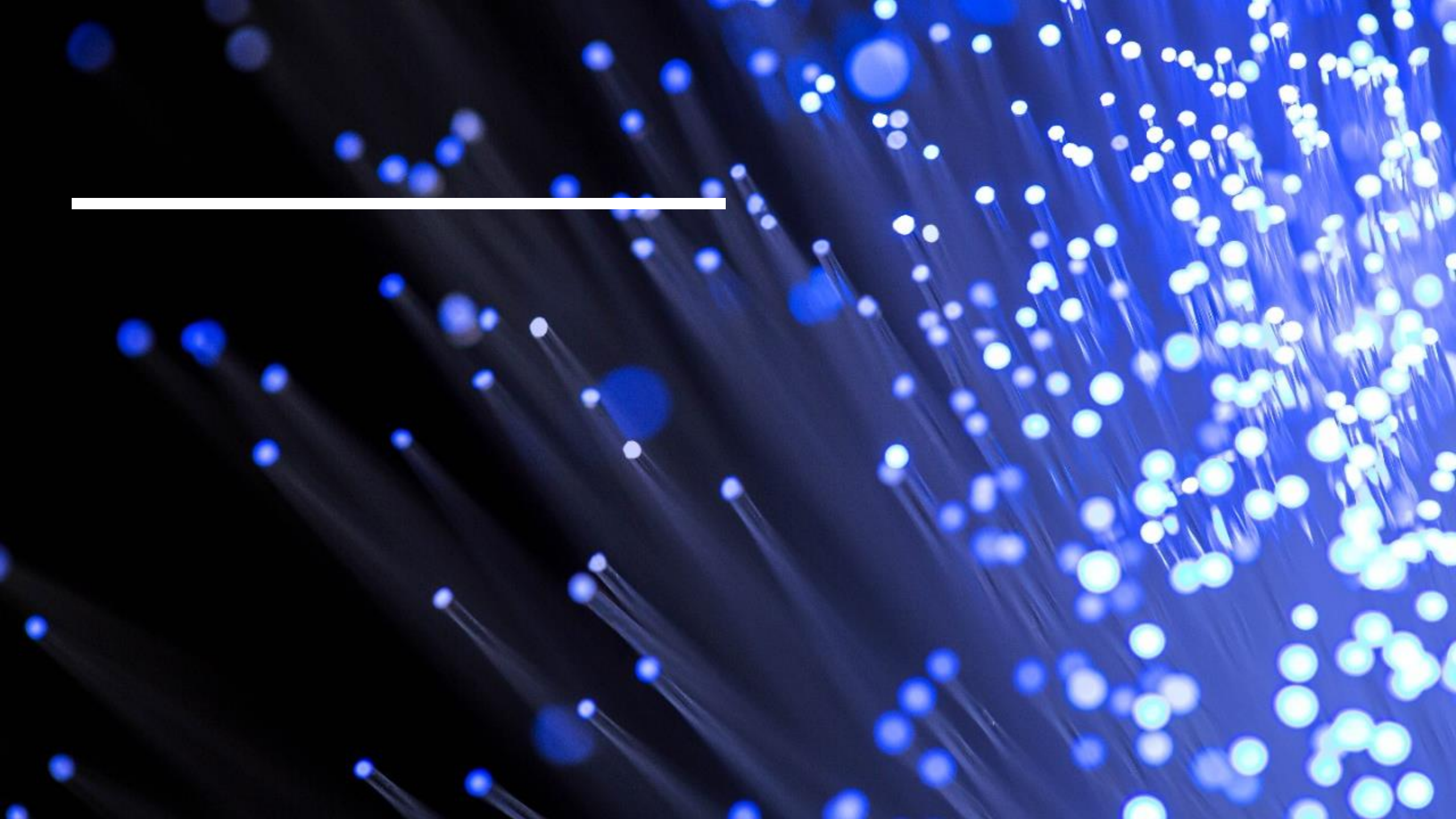
Frontend
Visuals

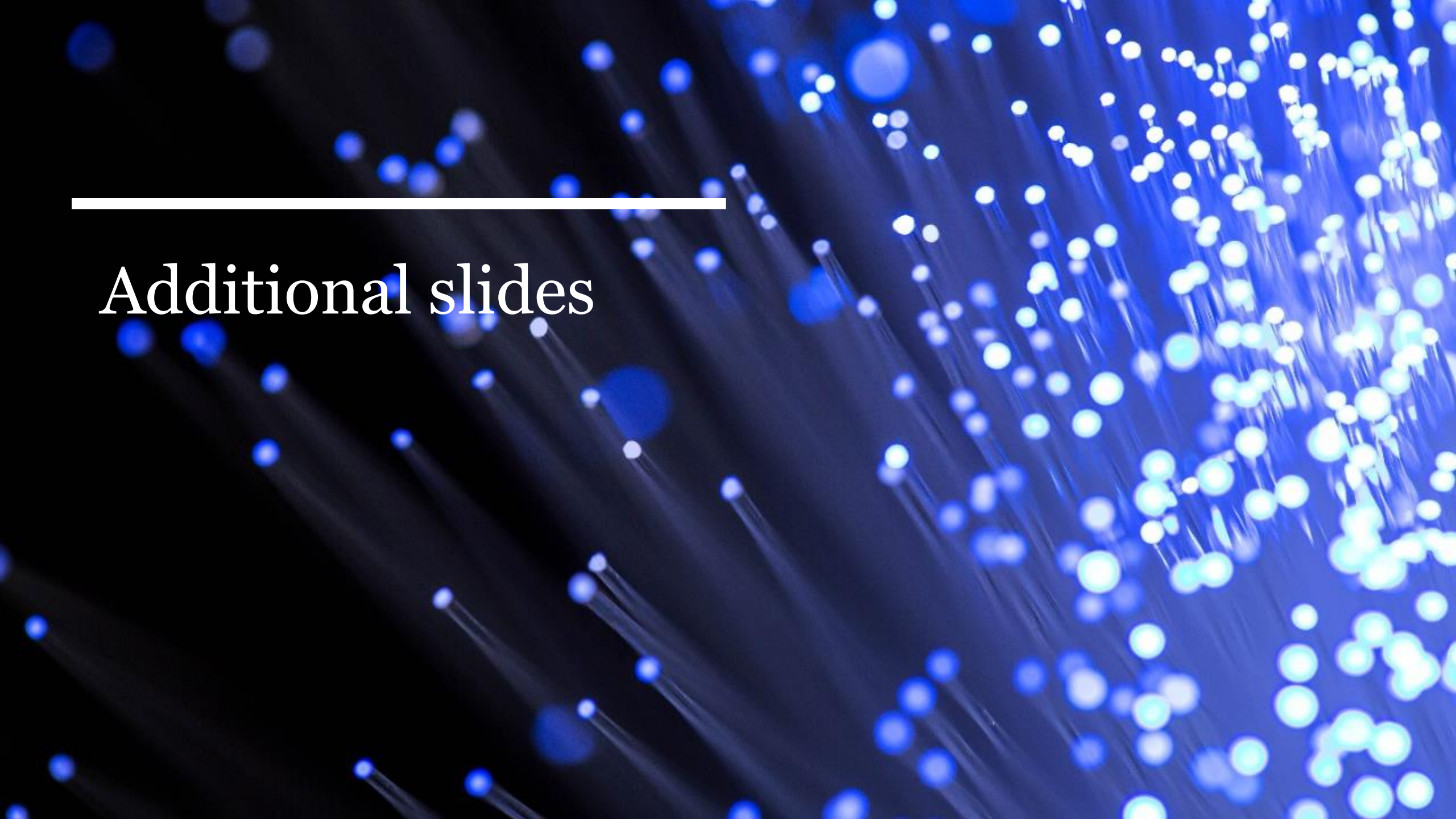
*This is an oversimplified overview. E.g. most often, a database is also implemented

As such, we'll build a stand-alone web application today

- No server side configuration
- Flask web application containing APIs
- Simple HTML page to display results
- Deploy to public url

- In short; A fully functional AI service





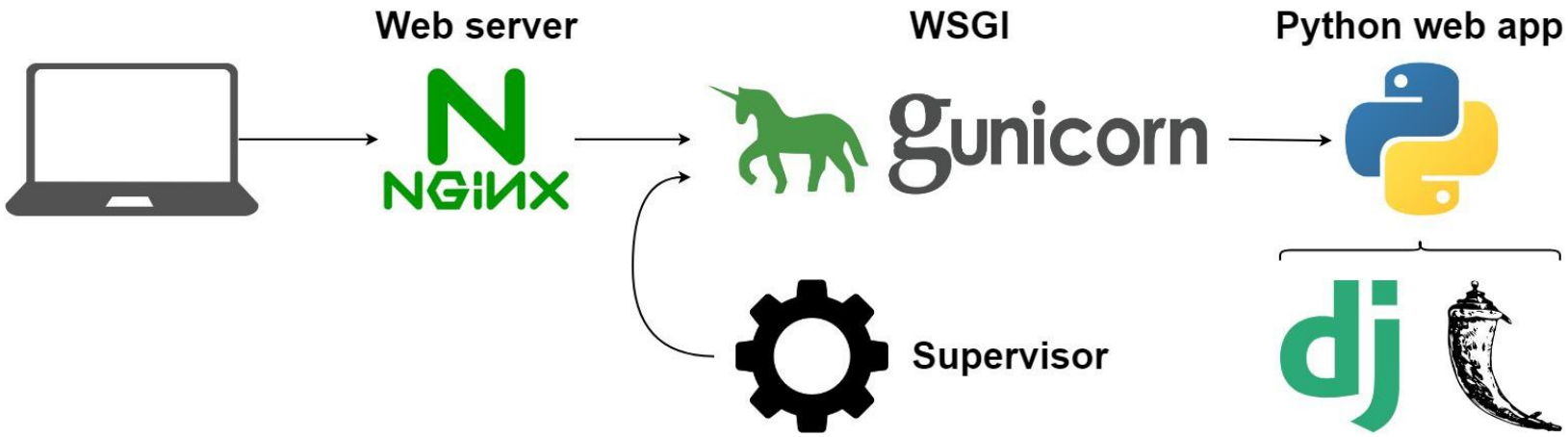
Additional slides

Deploying Flask to the public web

General roadmap

Create the content you want to serve (eg. Static ML model)	Set up server (eg. Python-Flask)	Open gateway for service consumption (eg. APIs)	Deploy to public url (eg. Heroku)

Flask specific



A Data Scientist should not be a Systems Engineer, although independent micro-apps are powerful

A more advanced example:

https://appliedcoding.net/apps/machine-learning-in-wine/wine_predictor/

Flask deployment to Heroku:

https://www.geeksforgeeks.org/deploy-python-flask-app-on-heroku/

Useful tools:

Git – virtual environments & requirements file – Command line understanding