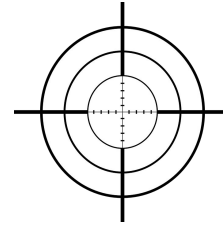
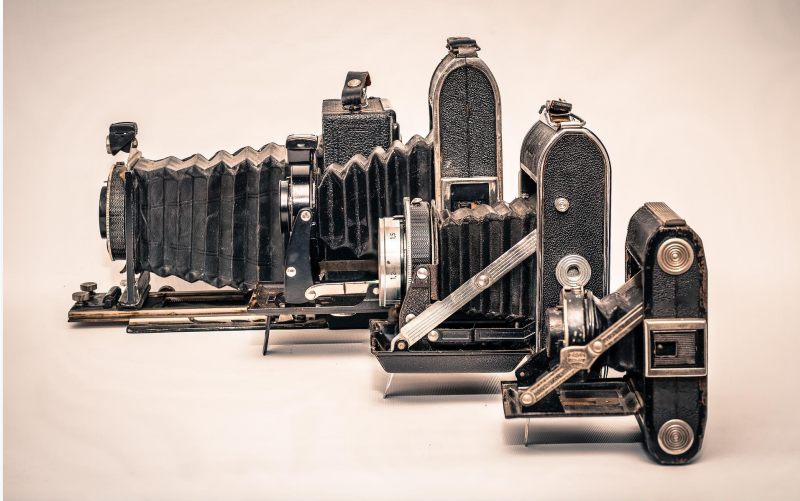


YOU HAVE BEEN HIRED BY The Sharp Shooting Company



Historical Business

Selling cameras , camcorders & accessories



New Business

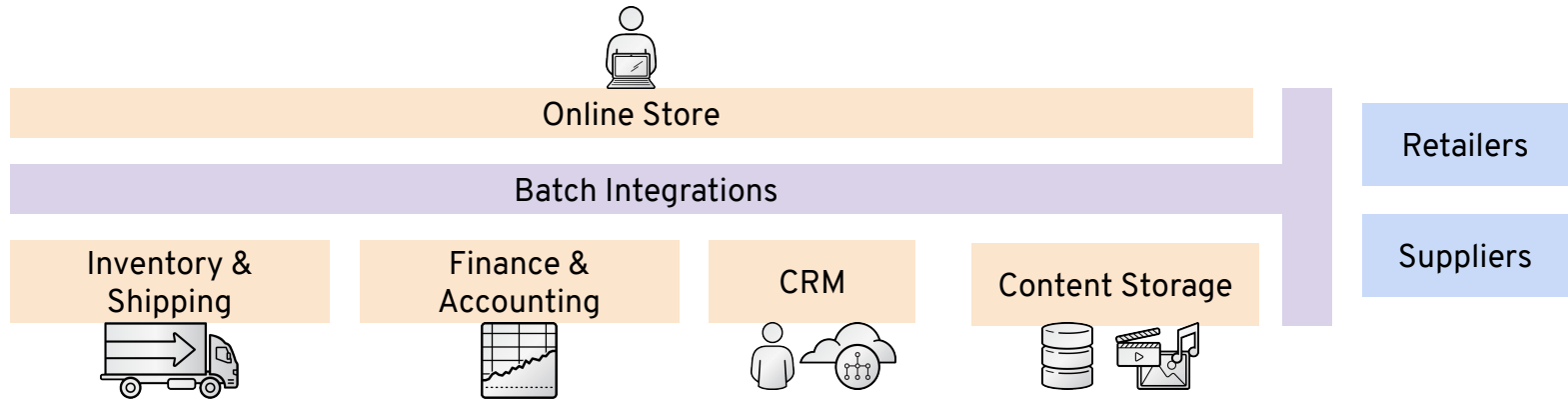
Selling affordable consumer drones with HD cameras



The challenge !

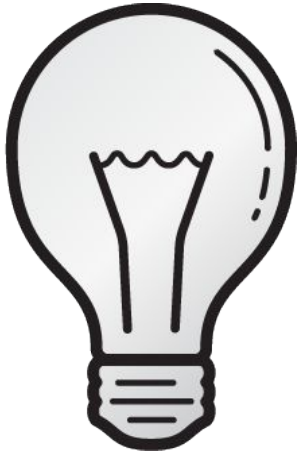
- Sales are less than expected for the drones !
- Main suspected reason :
 - Belief that beautiful drone shots are reserved for pros.
 - Fear of flying drones.
- How to get people more excited about great drone shots ?
- Constraints
 - Limited Marketing Budget, needs to be economically self sustainable.

Current Architecture



An example organisation

The Idea



Get great shots without owning a drone ?

The Thought Process : A Virtuous Circle

People see more beautiful drone footage

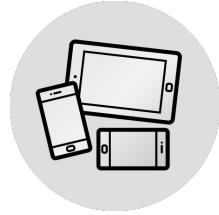


It becomes ubiquitous

People buy more drones

Lets create an App ?

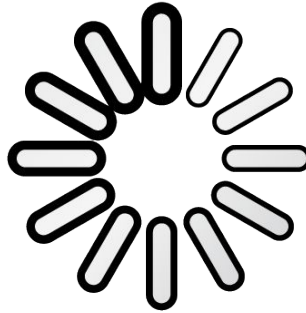
App that enables to buy existing drone shots, that can be easily shared on social media



OpenStreetMap API



Internal Process



Get the footage



The constraints

- Release a first version as fast as possible while staying future proof !
- Adopt a modular approach to maximise flexibility
- Leverage knowledge and capabilities from diverse teams
- Open up API to partners
- Ensure Scalability of each component
- Offer users/partners the ability to order custom drone footages
- Offer users the ability to vote for the next location to be added
- Take advantage of application user base to advertise drones

=> You are locked in this War Room until you find a viable solution

Your Architecture

OSM Api



External
Services

CRM



Finance &
Accounting

Content Storage

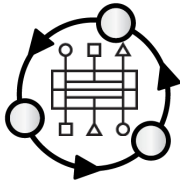
Backend Systems

Your Supporting Practices and
Methodologies

Security gateway



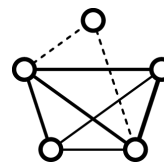
Api management



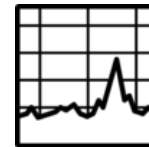
Flash Player



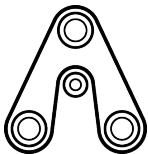
P2P



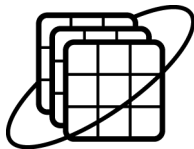
Business Activity Monitoring



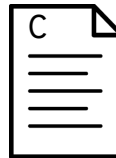
MicroService



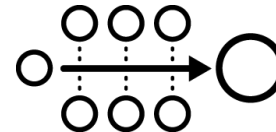
Composed Service



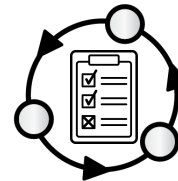
Cobol CopyBook



Business Process
Orchestration



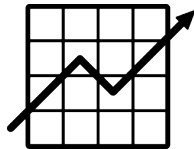
Decision Service



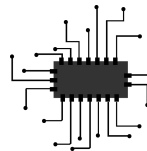
Distributed Cache



Planning Engine



Circuit Breaker



Ad Blocker



Dynamic Case Management



Messaging Broker



Serverless



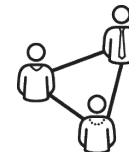
Knative



Container



DevOps



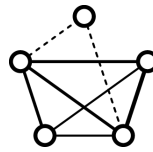
TDD



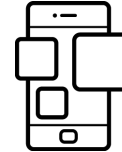
RPA



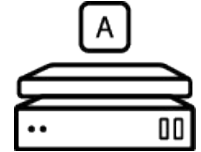
Blockchain



Mobile App



Agile Methodology



Progressive Web App



Antivirus

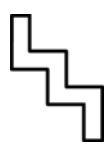


RUP

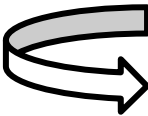
V-Model



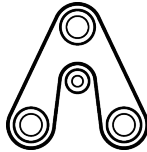
Waterfall



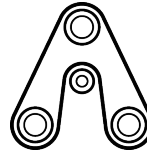
Data Replay Service



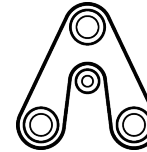
MicroService



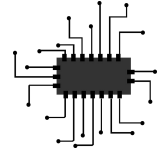
Throttler



Throttler



Circuit Breaker



Bonus Enigma :

The CRM is slow, can only process at max 5 transactions / seconds. Scaling up is not possible due to the legacy aspect. During busy moments there can be 100 of transactions per second. Integrating data later as a batch is acceptable.

What can be done to protect it from being overused?

What pattern can be used to store the excess data temporarily and process it by batch later.

