360° Webcams for Zoos and Aquariums

Team 12

Nathan Cool, Zach Newton, TJ Yacoub,
Sam Abdeltawab, Ian Jamieson, and Alan Negrete

Advisor: Dr. Henry Duwe

Client: Chris James (True 360)

sddec18-12.sd.ece.iastate.edu

Overview

- Project Plan
- Design Document
- Conclusion
- Q&A

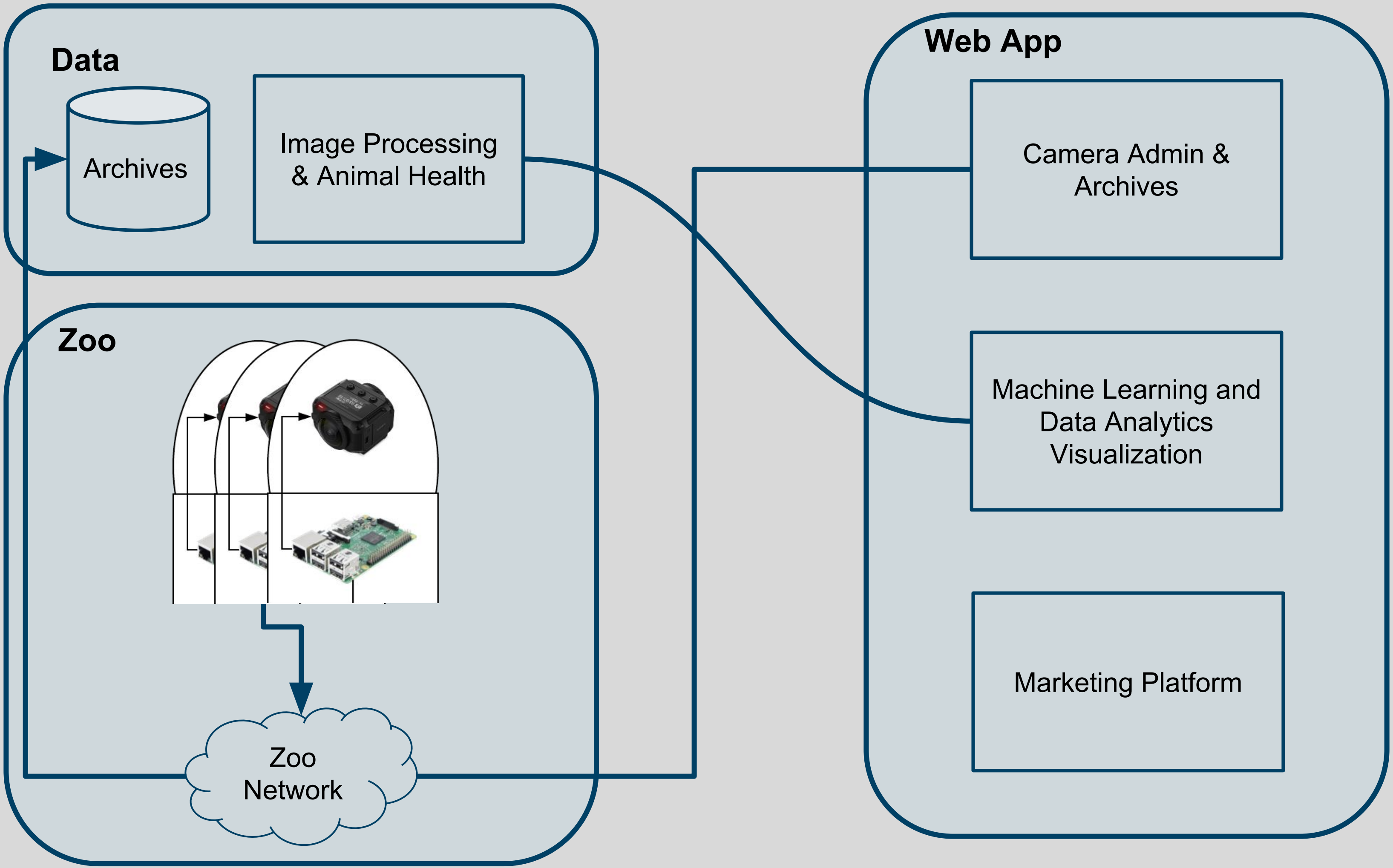
Project Plan

Problem Statement

- Poor video quality
- Footage is not interesting
- Animal may not be active
- Difficult to maintain

Our Solution

- Scalable SW/HW system
- Administrative portal
- Animal health monitoring
- Marketing tools


Data
Archives
Image Processing & Animal Health
Zoo
Zoo Network
Web App
Camera Admin & Archives
Machine Learning and Data Analytics Visualization
Marketing Platform

Functional Requirements

- 360° Camera System
  - Live stream up to 20 cameras at 4K resolution (30FPS and bitrate is 35 Mbps)
  - Capture photos in 4K resolutions
  - Archive past live streams for future viewing
- Administrative Portal
  - IT staff must be able to control/monitor the webcams
  - IT staff must be able to view archived live streams

Functional Requirements

- Marketing Tools
    - Marketing staff must be able to integrate logos into live streams
    - Sponsors can “rent” allotted times for selected streams to display their logos
- Animal Health Monitoring
    - Zoo curators must be able to monitor animal activity remotely
    - Zoo curators must be able to revisit video clips throughout the day
    - Send alerts to zoo curators when there are abnormalities

Non-Functional Requirements

- Cameras should be operable in many environments
- For database backups, Use AWS snapshot feature
- For scalability, the backend will rely on AWS
- For remote maintenance, Snappy will be used

Potential Risks & Mitigation

- Zoos do not have fast enough internet
- Camera system does not work underwater
- Zoos might want to use existing hardware (e.g., servers)
- Research efforts consume significant time
- Ethical issue of analyzing video clips with visitor’s faces

Market Survey

- While some zoos may use webcams, many lack 360° capability
- Camera setup and maintenance is tedious
- Zoos don’t have large IT teams.

Example: San Diego Zoo

- Low-quality/unstable live streams
- Located on zoo’s site rather than widely used platform (e.g., YouTube)
- Non-interactive and non-rewindable (static video)

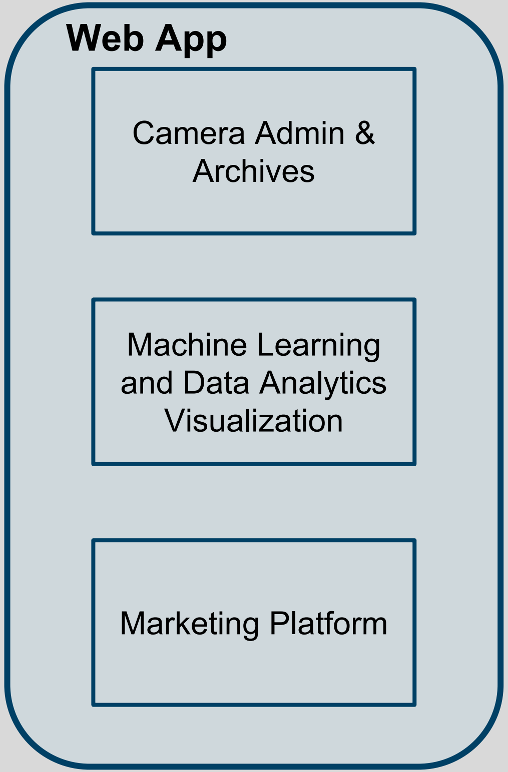
Resource/Cost Estimate

- Unit Cost
  - Camera: $800
  - Raspberry Pi board: $35
  - Cables, case, etc: $100
  - **Total = $935**
- Storage Costs
  - Server/storage: $1050 (3-5TB of data)
- Per-zoo example: Des Moines Zoo
  - Assume 10 cameras
  - (10 * unit cost) + per-zoo costs
  - **Total = ($935 * 10) + $1,050 = $10,400**

System Design

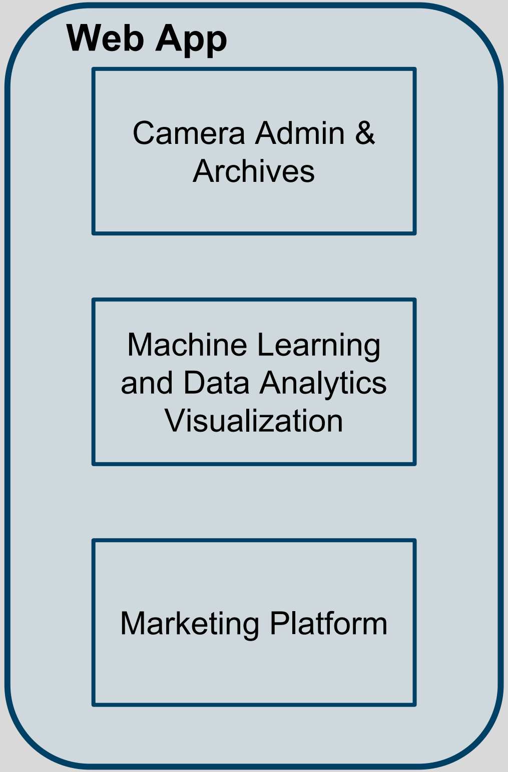
Detailed Design - Web App

- Cameras
  - View/edit camera data and settings
  - Control cameras (on/off, start/stop live streams, etc.)
- Archives
  - View/manage archived footage
- Technologies
  - Client: React.js + Redux
  - Server: Node.js
  - Authentication: Firebase



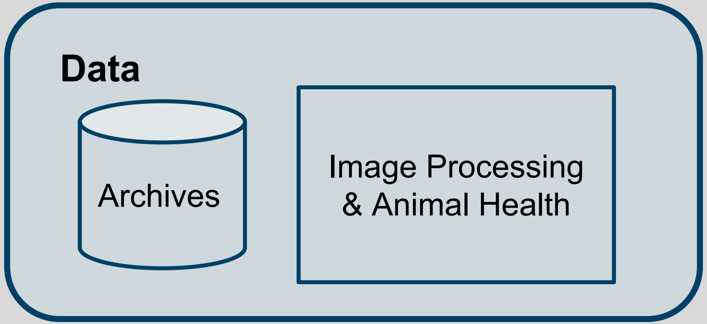
Detailed Design - Web App

- Animal Health Monitoring
  - Display animal activity data
  - Provide alerts of abnormalities to staff
- Marketing
  - Facilitate sponsorships
  - Extract footage of active animals
- Technologies:
  - Data Visualization: D3.js



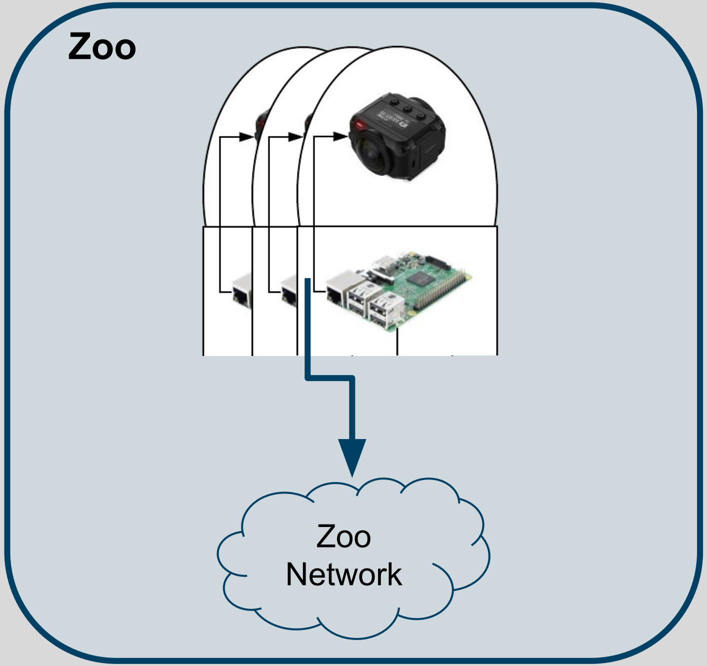
Detailed Design - Data

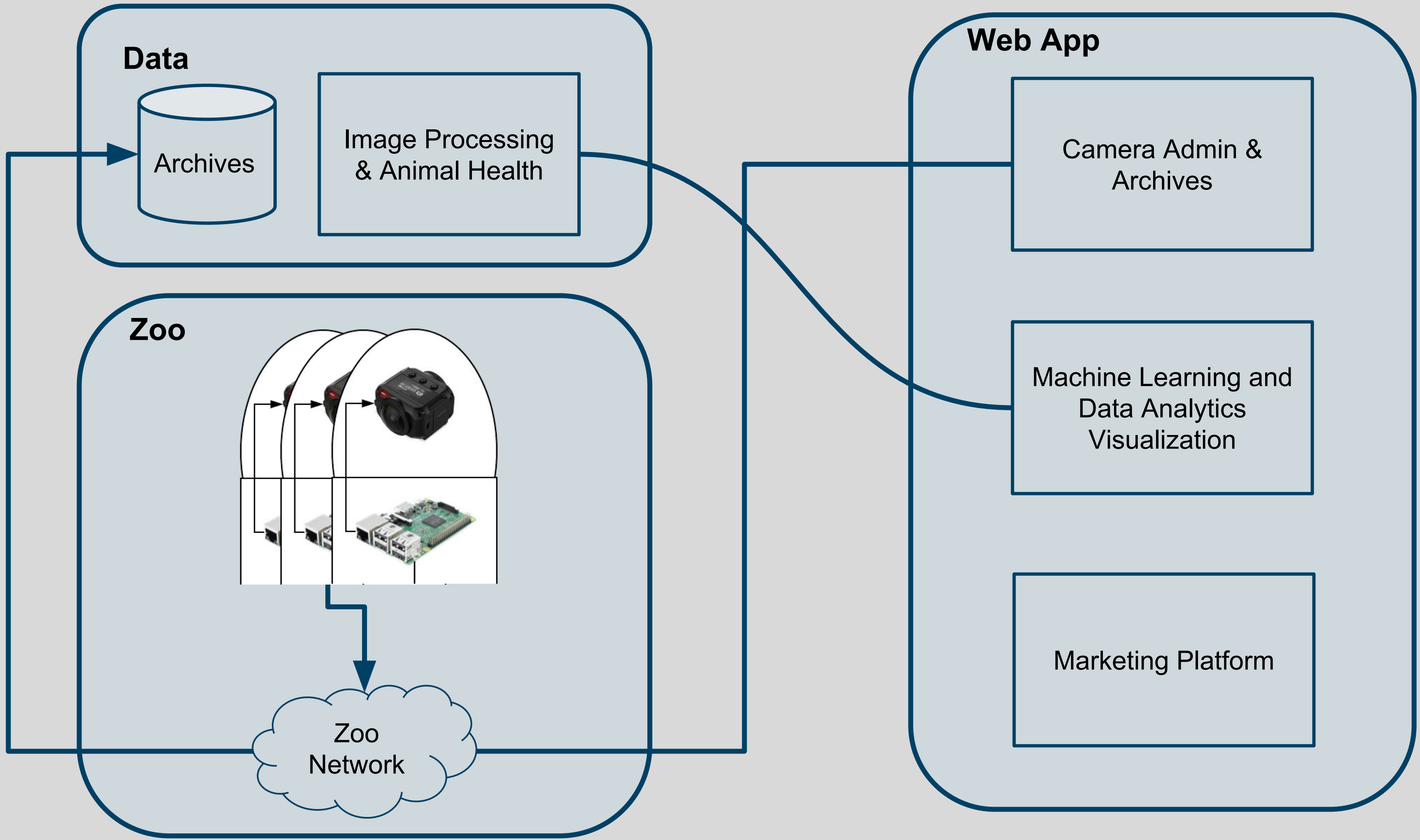
- Store video archives
- Analyze video archives
  - Image processing
  - Activity monitoring



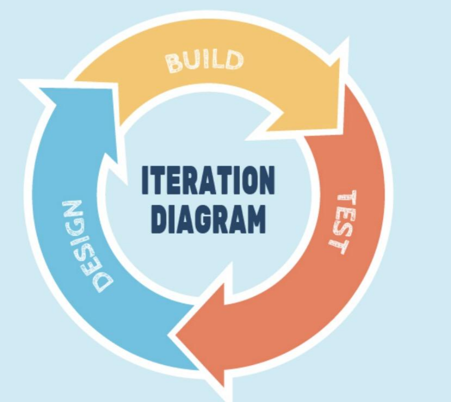
Detailed Design - Zoo

- Can contain one or many cameras
- Each camera controller is subscribed to its own channel
- Devices
  - Garmin Virb 360° (Works on all OS’s)
  - Raspberry Pi: Runs ARM OS’s
- Technologies
  - Python, FFmpeg, Pusher client




Data
Archives
Image Processing & Animal Health
Zoo
Zoo Network
Web App
Camera Admin & Archives
Machine Learning and Data Analytics Visualization
Marketing Platform

Iterative Test Plan



- Step 1: Design & Develop a component
- Step 2: Independently test component
- Step 3: Integrate and test with existing system
- Step 4: Test components against functional/non-functional requirements
- Step 5: Full system tests to evaluate system stress tolerance

Test Plan - Example

Archiving Video Test

- Goal: Meet functional requirement “IT staff must be able to view archived live streams”
- Process:
    1. Live stream and archive video for 8 hours
    2. Ensure all video files have been stored in server at zoo
    3. Check if all video files are accessible from the Web App
    4. Check if a specific date and time is accessible
    5. Verification

Prototype Implementations

- Earlier devices/prototypes
  - Poor SDKs & Developer support
  - Could not meet non functional requirements
- Current Prototype & Meeting Requirements
  - External hardware to communicate with camera
  - Easily set up & interact with minimum maintenance

Conclusion

Project Milestones & Schedule (Spring 2018)

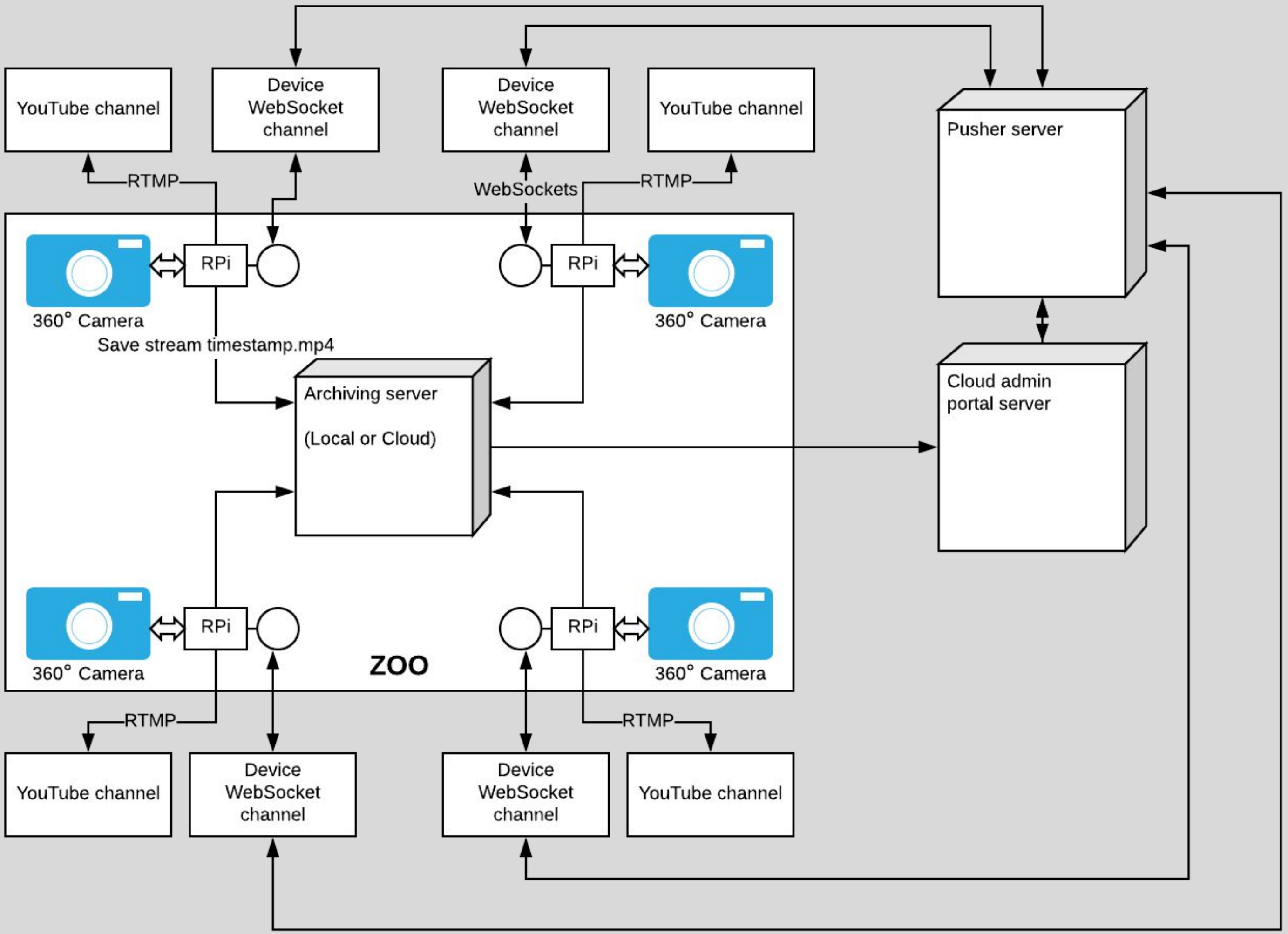
- Identify hardware/software and design system *(done: Feb. 28)*
- Develop software to connect webcams to YouTube *(done: Mar. 15)*
- Develop software to remotely control webcams *(done: Apr. 15)*
- Develop software to archive live video streams *(in progress)*

Project Milestones & Schedule (Fall 2018)

- Develop software which allows logos and advertisements to be embedded in a live stream *(due: Sep. 30)*
- Develop software which uses machine learning to observe animal health and behavior *(due: Nov. 22)*
- Develop software which assists in curating potential marketing materials from existing video archives *(due: Nov. 22)*
- Deploy the system at a zoo/aquarium *(due: Nov. 22)*

Q&A

Thank you!



YouTube channel
Device WebSocket channel
Device WebSocket channel
YouTube channel
Pusher server
RTMP
WebSockets
RTMP
RPi
RPi
360° Camera
360° Camera
Save stream timestamp.mp4
Archiving server
(Local or Cloud)
Cloud admin portal server
RPi
RPi
360° Camera
ZOO
360° Camera
RTMP
RTMP
YouTube channel
Device WebSocket channel
Device WebSocket channel
YouTube channel