

Next-generation Annotation Management

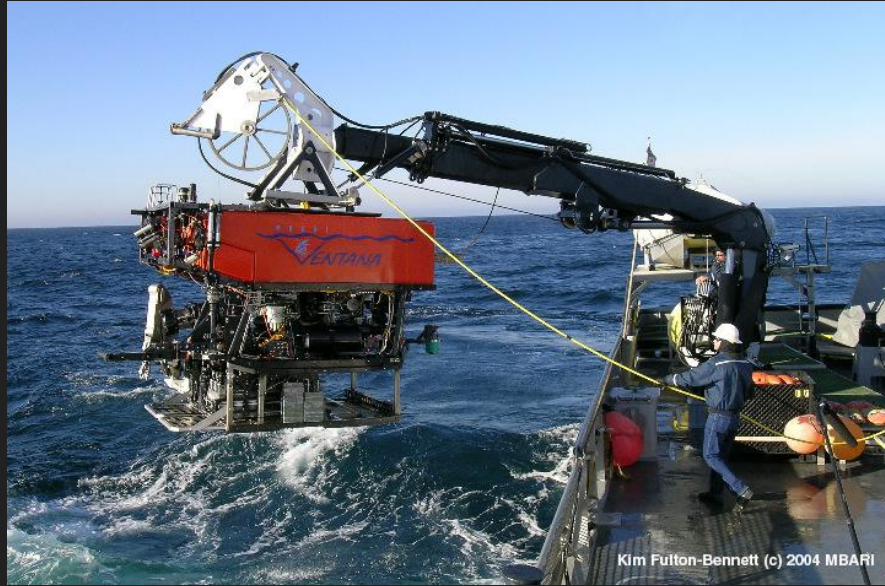
Brian Schlining



MBARI

Monterey Bay Aquarium
Research Institute

A Little History



A Little History

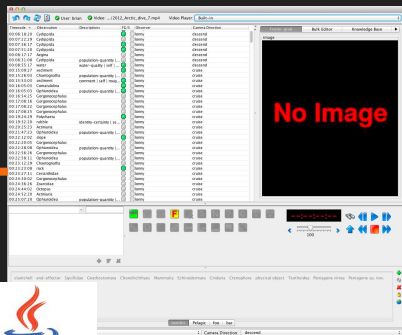
circa 2005



RDBMS



Cross-platform



Video
Annotation and
Reference
System

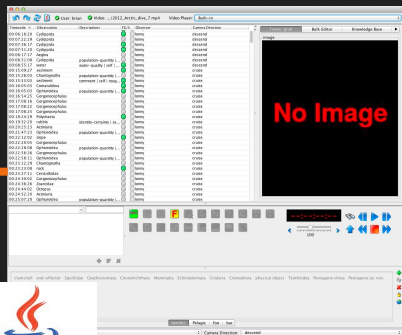
A Little History

circa 2005

VARS was designed for ...



RDBMS



Cross-platform



Frame capture hardware



Deck control via RS422



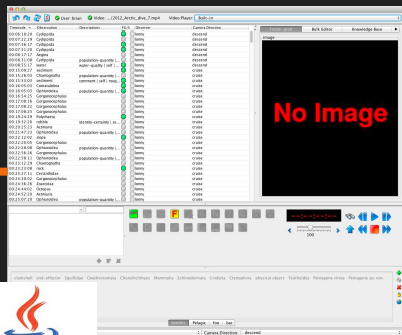
videotape

VARS Today

Today VARS is used for ..
with some caveats



RDBMS



Cross-platform



videotape



Video Files

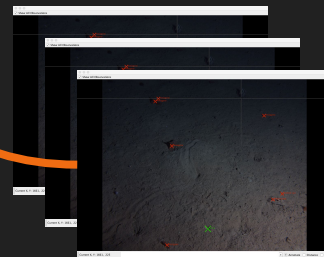
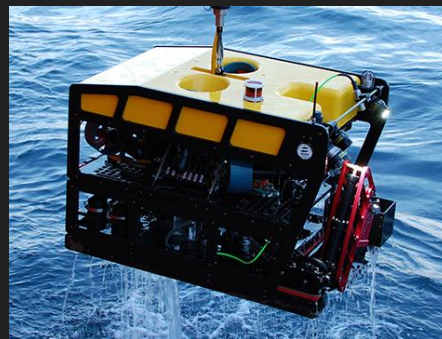
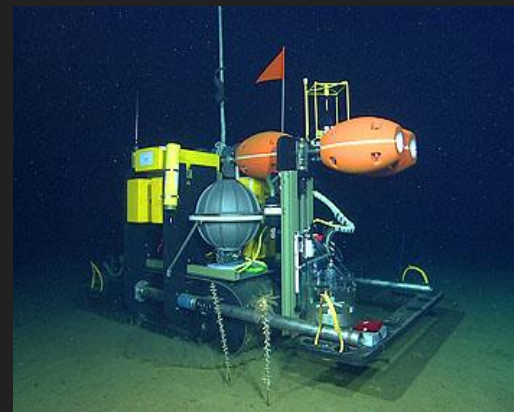


Image Collections

Image collection at MBARI



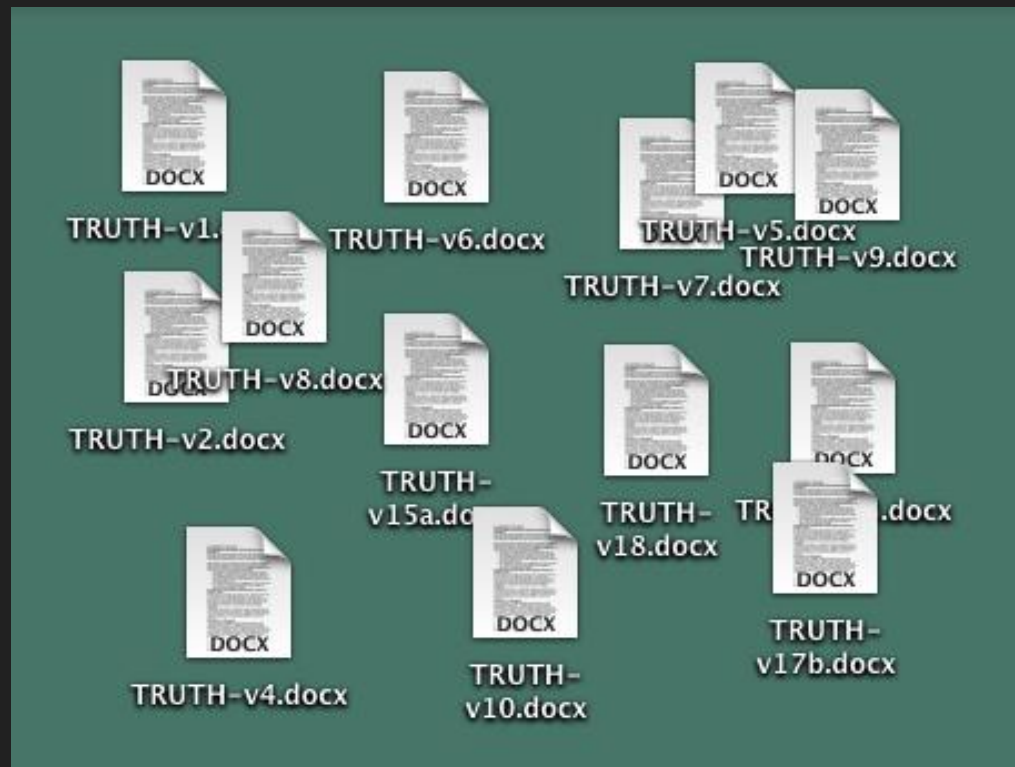
Lessons Learned

- Spelling Matters
 - Constrained vocabulary required!



Lessons Learned

- Spelling Matters
 - Constrained vocabulary required!
- Database >>> Files
 - No files to manage
 - Data always in sync
 - Easy to share data



Lessons Learned

- Spelling Matters
 - Constrained vocabulary required!
- Database >>> Files
 - No files to manage
 - Data always in sync
 - Easy to share data
- ~~One tool to rule them all~~
 - Researchers have different needs



Lessons Learned

- Spelling Matters
 - Constrained vocabulary required!
- Database >>> Files
 - No files to manage
 - Data always in sync
 - Easy to share data
- ~~One tool to rule them all~~
 - Researchers have different needs
- Power users need power tools



Lessons Learned

- Spelling Matters
 - Constrained vocabulary required!
- Database >>> Files
 - No files to manage
 - Data always in sync
 - Easy to share data
- ~~One tool to rule them all~~
 - Researchers have different needs
- Power users need power tools
- Digital video is hard
 - Many formats
 - Large file sizes
 - Many, many files to track



Lessons Learned

- Spelling Matters
 - Constrained vocabulary required!
- Database >>> Files
 - No files to manage
 - Data always in sync
 - Easy to share data
- ~~One tool to rule them all~~
 - Researchers have different needs
- Power users need power tools
- Digital video is hard
 - Many formats
 - Large file sizes
 - Many, many files to track



Software - Design Requirements

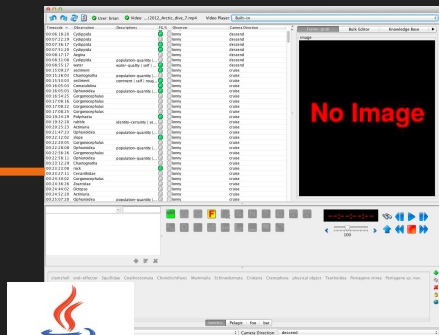
Goal: Continue to enable scientists to use MBARI collected video for qualitative and quantitative science.

1. Accommodate future changes to digital video (future-proofing).
2. Efficiently deliver video.
3. Track the locations of videos.
4. Manage videos annotations.

From This ...



RDBMS



Cross-platform



videotape



Video Files

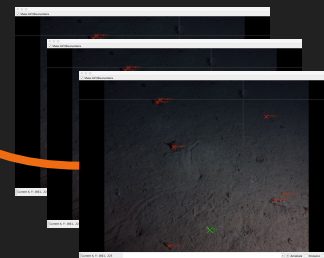
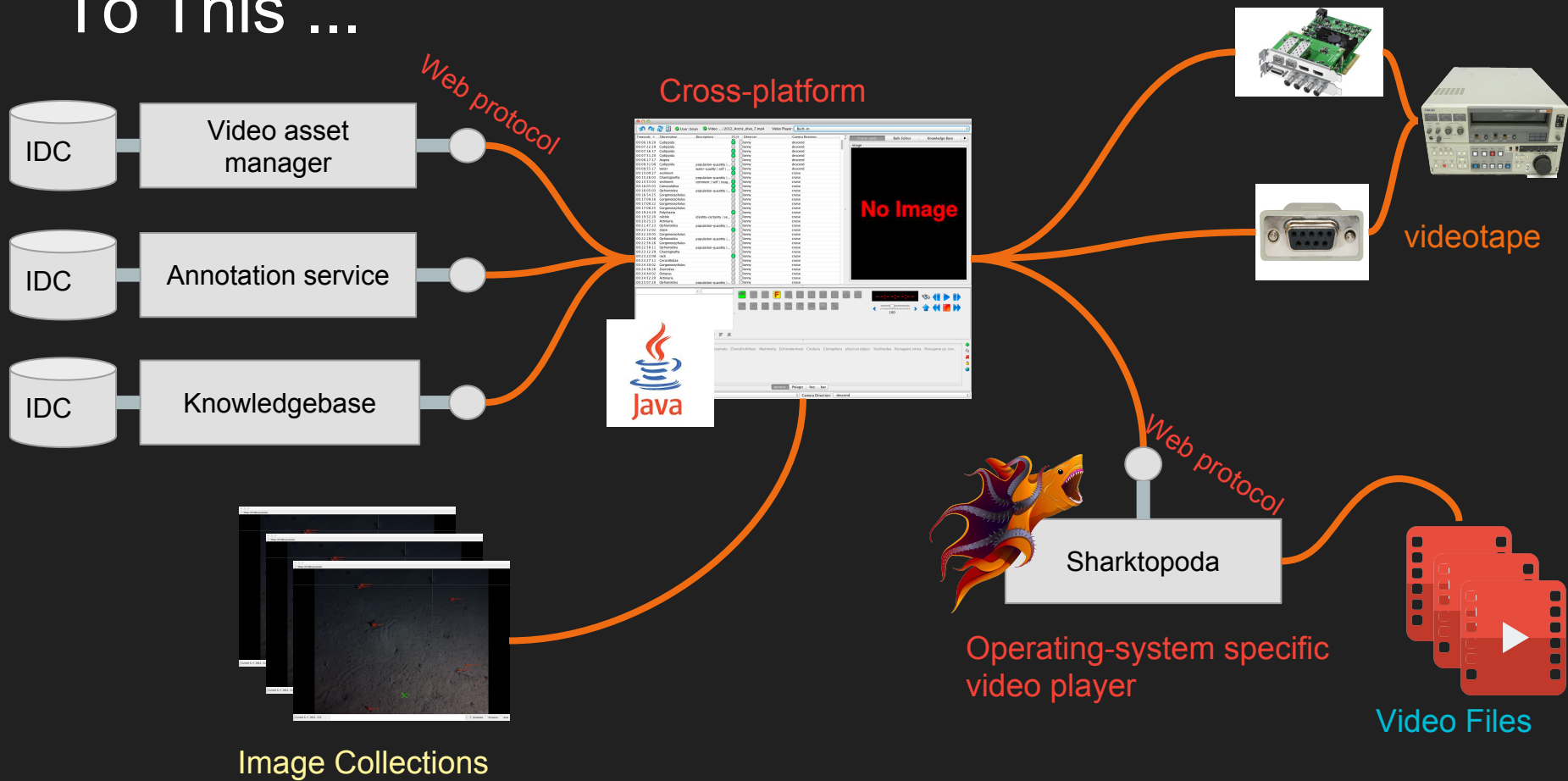


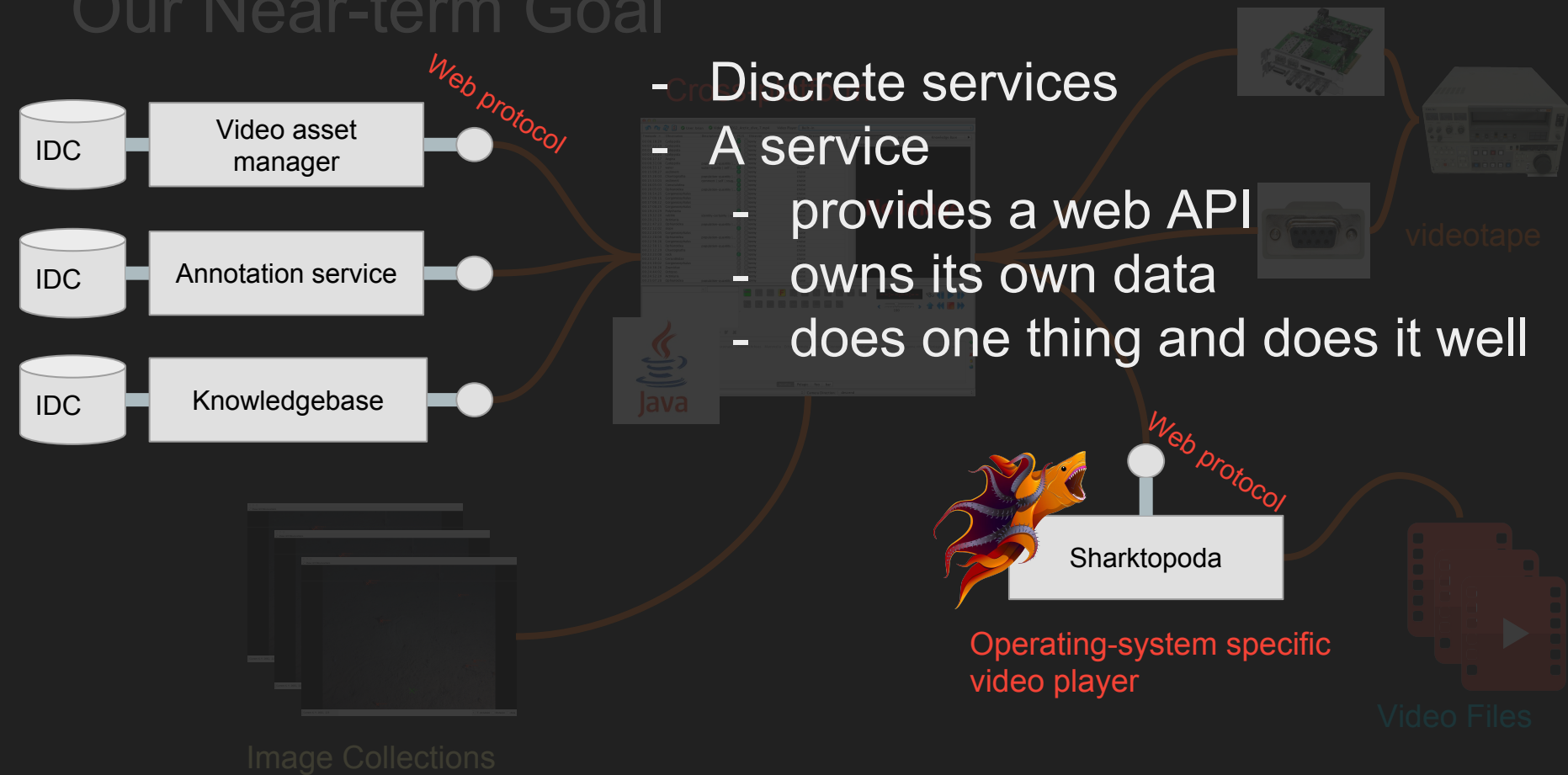
Image Collections

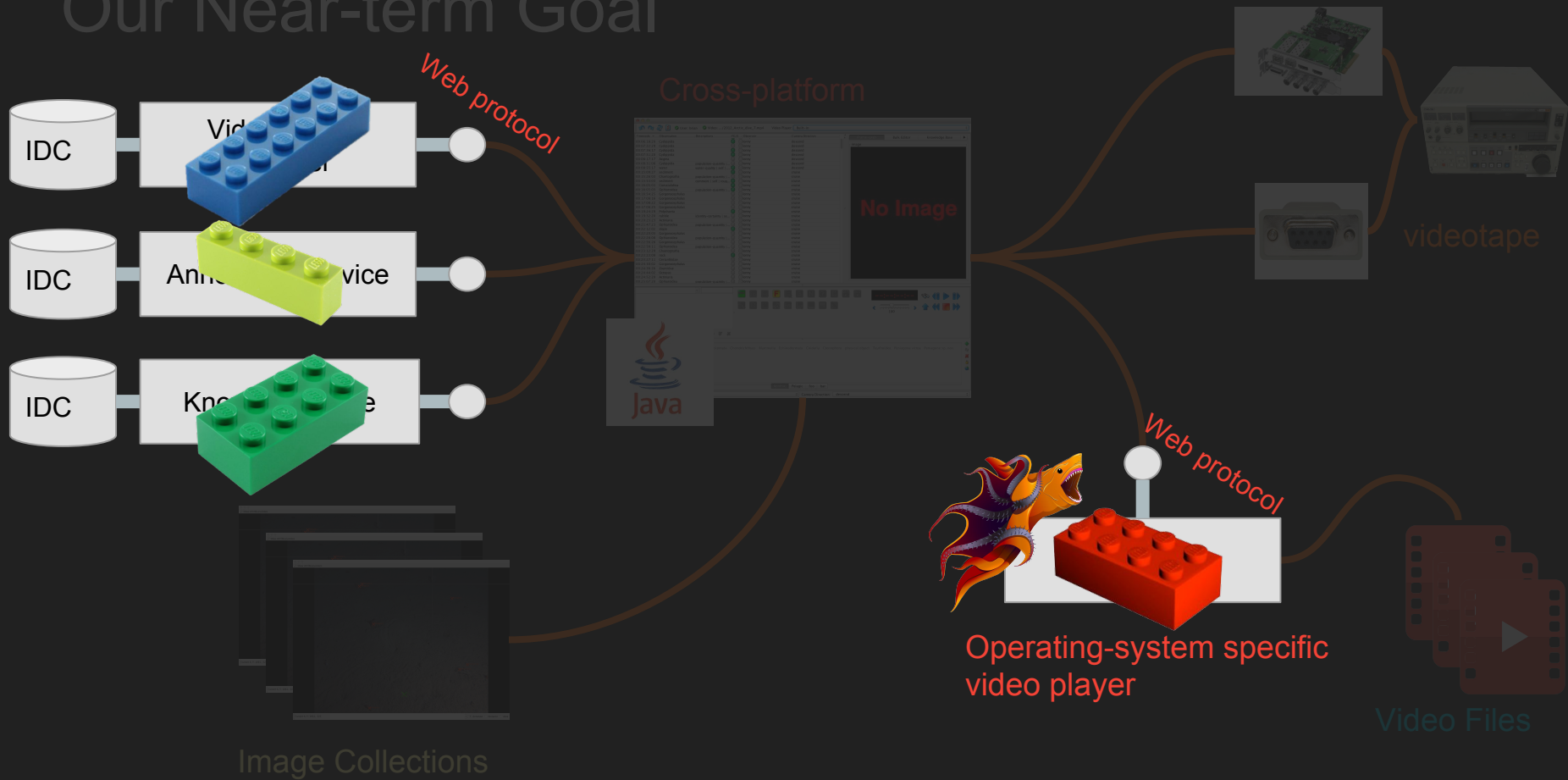
To This ...



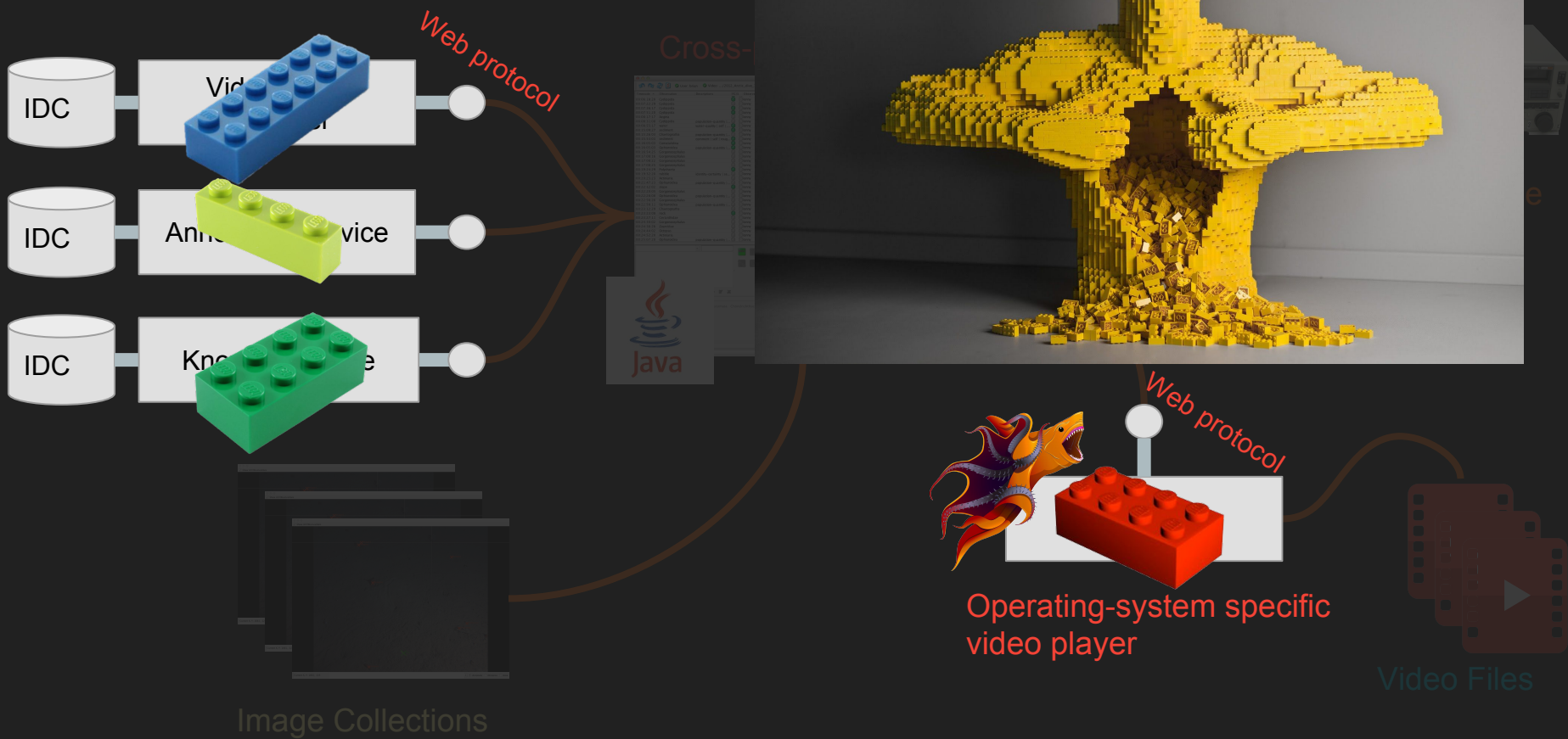
Our Near-term Goal

- Discrete services
- A service
 - provides a web API
 - owns its own data
 - does one thing and does it well





Our Near-term Goal



Our Near-term Goal

The architecture consists of the following components and data flow:

- Input Data:** Three IDCs (Image Data Collections) feed into three services:
 - Video asset manager
 - Annotation service
 - Knowledgebase
- Communication Protocols:**
 - REST/JSON:** Used for communication between the IDCs and the services.
 - Web protocol:** Used for communication between the services and the applications.
 - UDP/JSON:** Used for communication between the applications and the video player.
- Applications:** Multiple applications (represented by icons of a laptop, a server, and a mobile device) use the services. They are **Cross-platform** and **Programming language agnostic**.
 - One application is specifically labeled **Java**.
- Output:** The applications feed into the **Sharktopoda** component, which is an **Operating-system specific video player**.
 - The video player outputs **Video Files**.

- Apps use only services needed
- Programming language agnostic

UDP/JSON

REST/JSON

Sharktopoda

Operating-system specific video player

Using a Service - Python

```
import requests
import json
r = requests.get('http://foo.org/kb/v1/concept/Nanomia')
j = json.loads(r.text)
```

```
j['name']
j['alternateNames']
j['rank']
j['media']
```

Using a Service - Matlab

```
j = webread('http://foo.org/kb/v1/concept/Nanomia')
```

```
j.description
```

```
j.alternateNames
```

```
j.rank
```

```
j.media
```

Using a Service - R

```
install.packages('rjson')  
library('rjson')  
j <- fromJSON(readLines('http://foo.org/kb/v1/concept/Nanomia'))
```

```
j$name  
j$alternateNames  
j$rank  
j$media
```

Using a Service - Perl

```
$j==0&&($C++,$C>=$a&&($C=0));$j==2&&($C||($C=$a),$C--);  
$j==3&&($R++,$R>=@B&&($R=0));$j==1&&($R||($R=@B),$R--);}   
KP($){push@S,shift}KJ(){pop@S||0}KX(){@S[-1,-2]=@S[-2,-1]}KR(  
{push@S,$S[-1]}KW($$){"Z".$_[0]."Z,K{".$_[1]."},"}KG($){($_)=($a)=@_  
y/\`/>/;W$a," X;P(J$_ J)"}KD($){($a)=@_;W$a,ZP Z.$a}KE($){($_)=($a)  
=@_;y/0123/>^<v/;W $_,"N $a"};y/Z/\';s/K/s$I /g;my($R,$C,@S);  
eval$_;}$_=$x=W ' ','N(J?2:0)';y/_02/|31/;$x.$_  
#r^>\"J eg"1+T,,,l#
```


Our Near-term Goal

vampire-squid

annosaurus

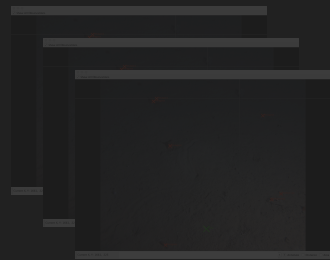
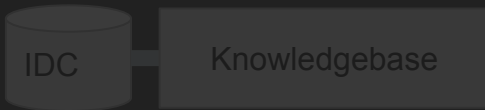
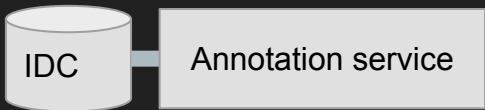
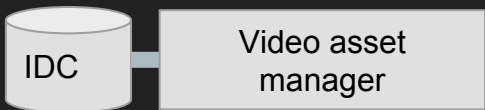
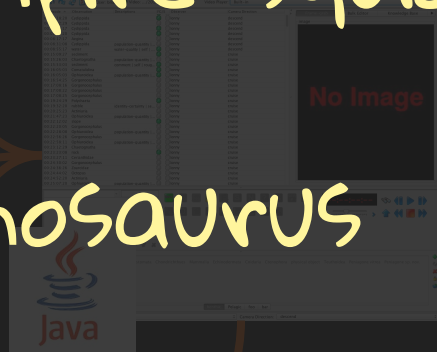
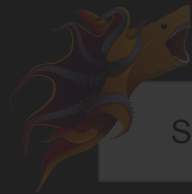


Image Collections

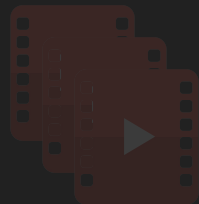


videotape



Sharktopoda

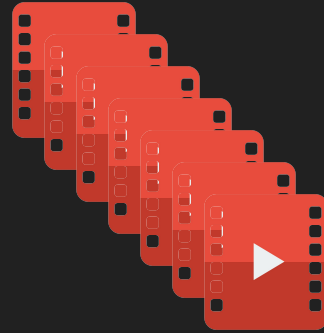
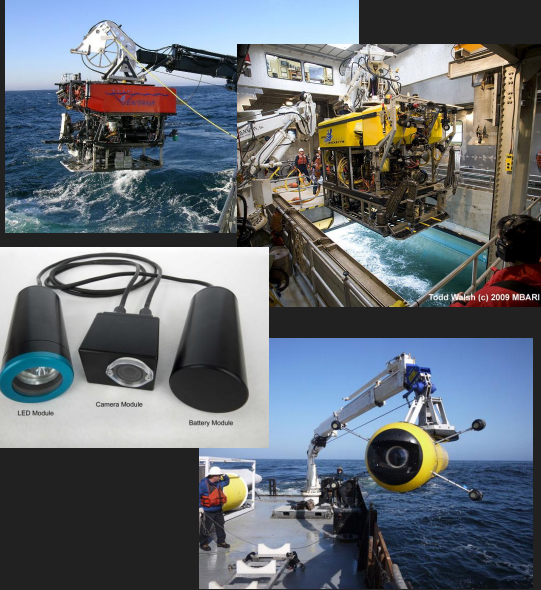
Operating-system specific
video player



Video Files

Vampire-Squid

Efficiently Deliver Video to Users - Data Collection



Deployment

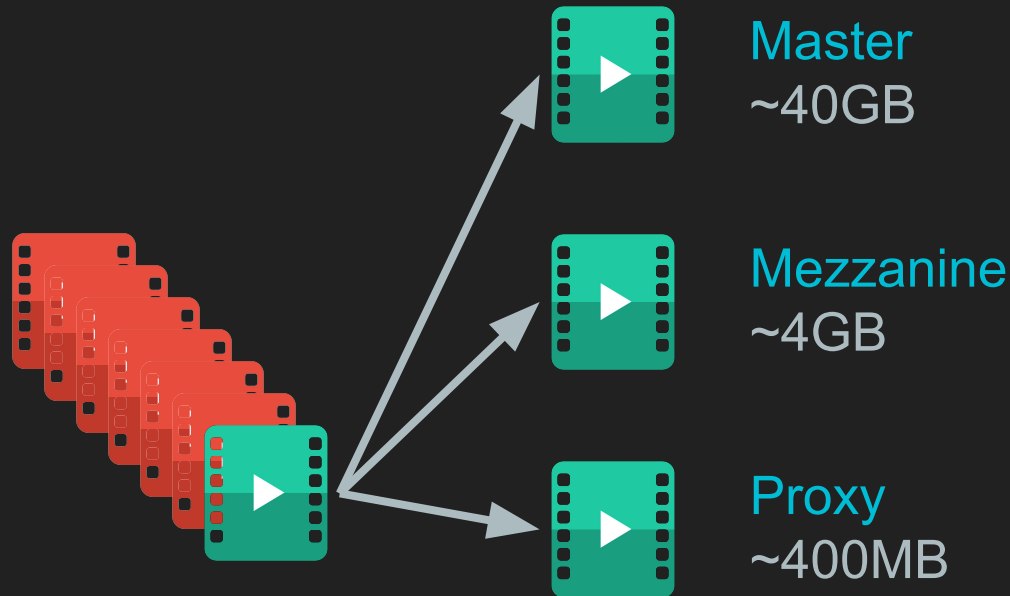
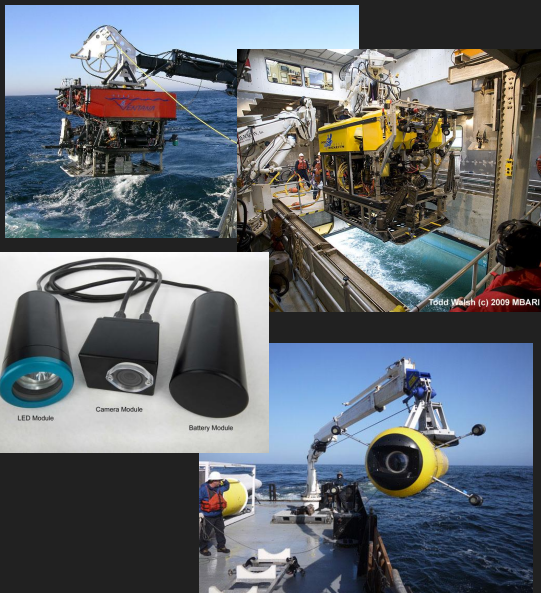
8 hour dive is ~1TB

Video Segments

32x 15-min videos

Vampire-Squid

Efficiently Deliver Video to Users - Data Collection



Deployment

8 hour dive is ~1TB

Video Segments

32x 15-min videos

Efficiently Deliver Video to Users - Data Archiving

- Capture resolution
- Minimal compression
- For detailed annotations



Master
~40GB



Mezzanine
~4GB



Proxy
~400MB

Efficiently Deliver Video to Users - Data Archiving

- Capture resolution
- Minimal compression
- For detailed annotations
- Compressed
- For generating proxies



Master
~40GB



Mezzanine
~4GB



Proxy
~400MB

Efficiently Deliver Video to Users - Data Archiving

- Capture resolution
- Minimal compression
- For detailed annotations
- Compressed
- For generating proxies
- Highly compressed at various resolutions
- For outline annotations



Master
~40GB



Mezzanine
~4GB



Proxy
~400MB

Efficiently Deliver Video to Users - Data Delivery

- Physical delivery mechanism
- Prefetch via web



Master
~40GB

- Streaming via web



Mezzanine
~4GB

- Streaming via web



Proxy
~400MB

vampire-squid

96 videos / deployment

X

300 deployments /year

~30,000 videos / year

Vampire-Squid

Allows Applications to Ask:

96 videos / deployment

X

300 deployments /year

~30,000 videos / year

For a given deployment and/or moment in time:

- What videos are available?
- Where are they?
- What are their:
 - codecs
 - containers
 - resolution
 - framerate
 - fingerprint (SHA512)

vampire-squid

vampire-squid

A simple custom video asset manager.

- <https://github.com/underwatervideo/vampire-squid>
- Open source

```
docker run -p 8080:8080 hononuu1i/vampire-squid
```

Annotation API

annosaurus

Goal: Make it easy for researchers to create and edit annotations from their own applications

- An evolved VARS annotation data model.
- Rich, flexible data model

Annotation API

annosaurus

Goal: Make it easy for researchers to create and edit annotations from their own applications

- An evolved VARS annotation data model.
- Rich, flexible data model

Want to search for any kind of squid eating any kind of Myctophidae between 200 and 400 meters in June 2007?

Image Annotation Supported

annosaurus

Index is timestamp



2017-02-21T16:11:09Z



Image Annotation Supported

annosaurus

Index is timestamp



2017-02-21T16:11:09Z



Can have multiple images at one index. e.g.

- left/right stereo image
- raw/color corrected image

Annotations apply across all versions of a video

annosaurus + vampire-squid

← Index into Media (Elapsed-time and/or Timecode)

3943735 millis / 01:05:43:22



Master

Mezzanine

Proxy

Annotation - Software Infrastructure

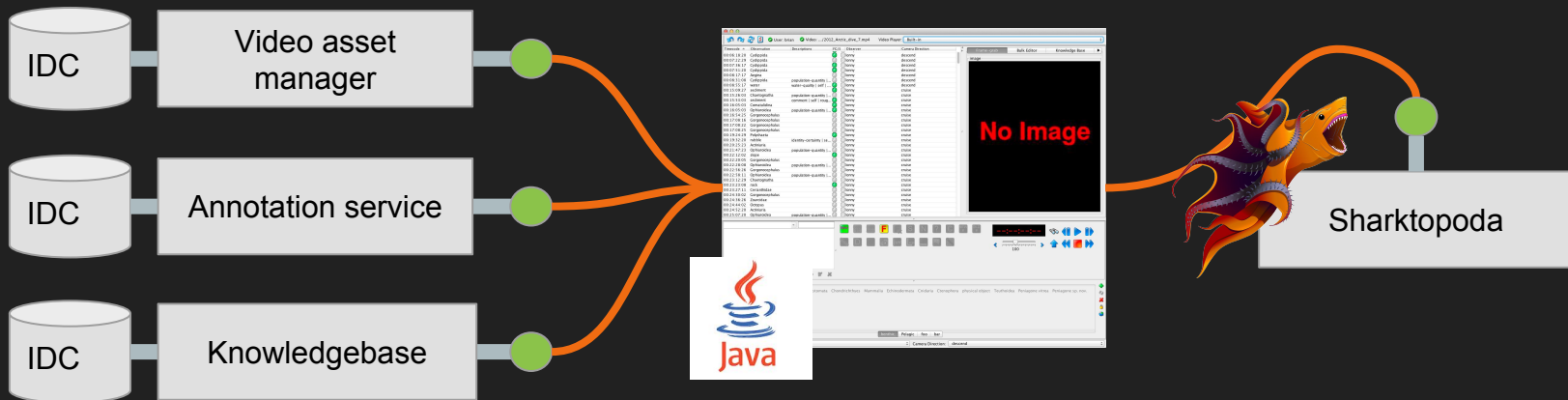
Video and Image annotation service

Annosaurus

- <https://github.com/underwatervideo/annosaurus>
- Open source

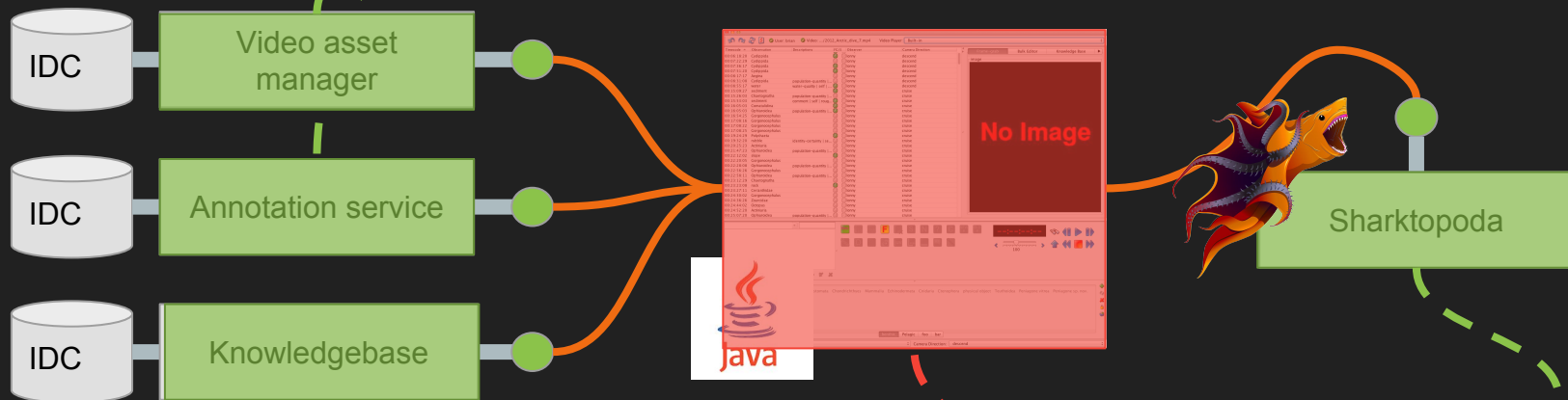
```
docker run -p 8080:8080 hohonuuli/annosaurus
```

Status



Status

Alpha - ready for integration



Completed

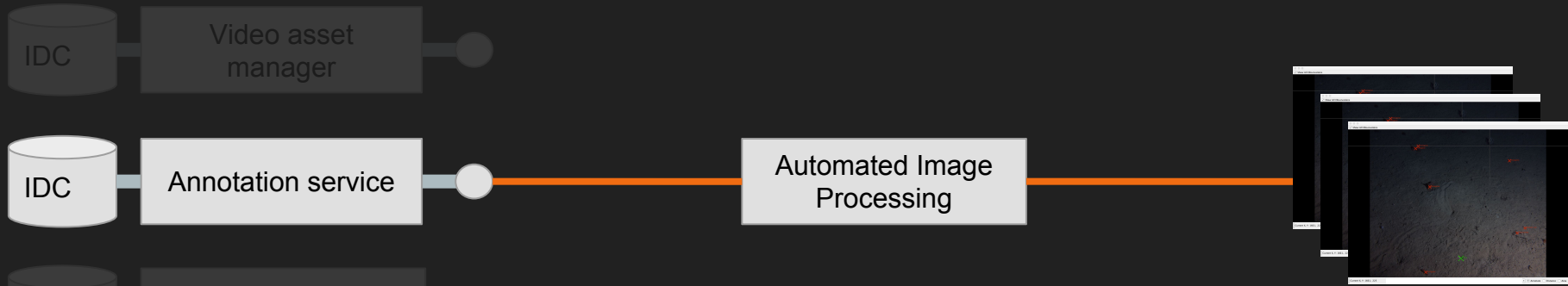
In Production

Web API is Alpha

Rewrite required

Existing app to be rewritten to new APIs

Potential Applications - Automation

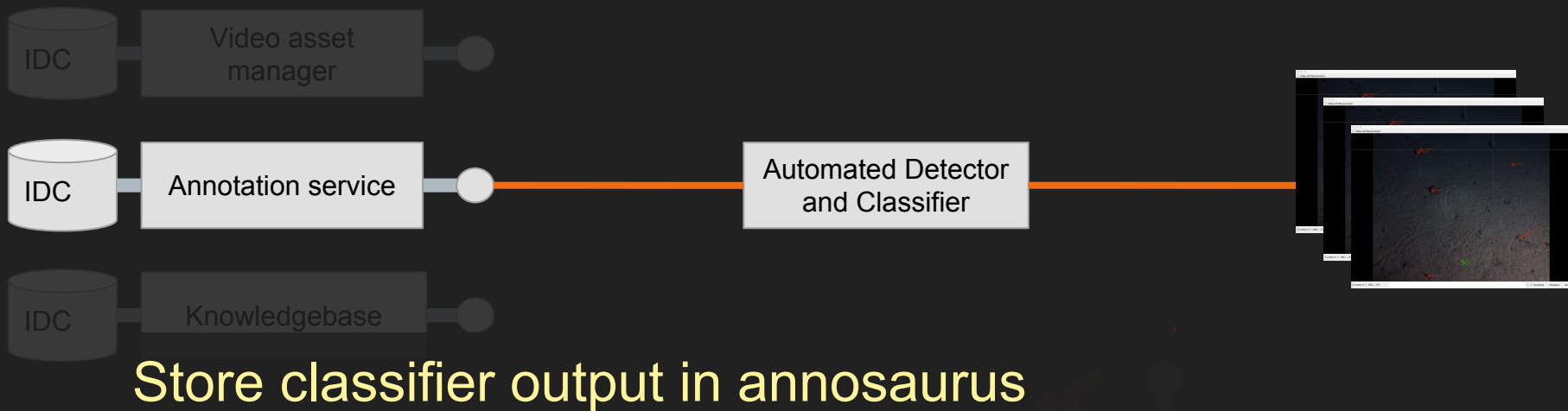


Color correct, remove marine snow etc.

Register corrected images with annosaurus

video player

Potential Applications - Automation



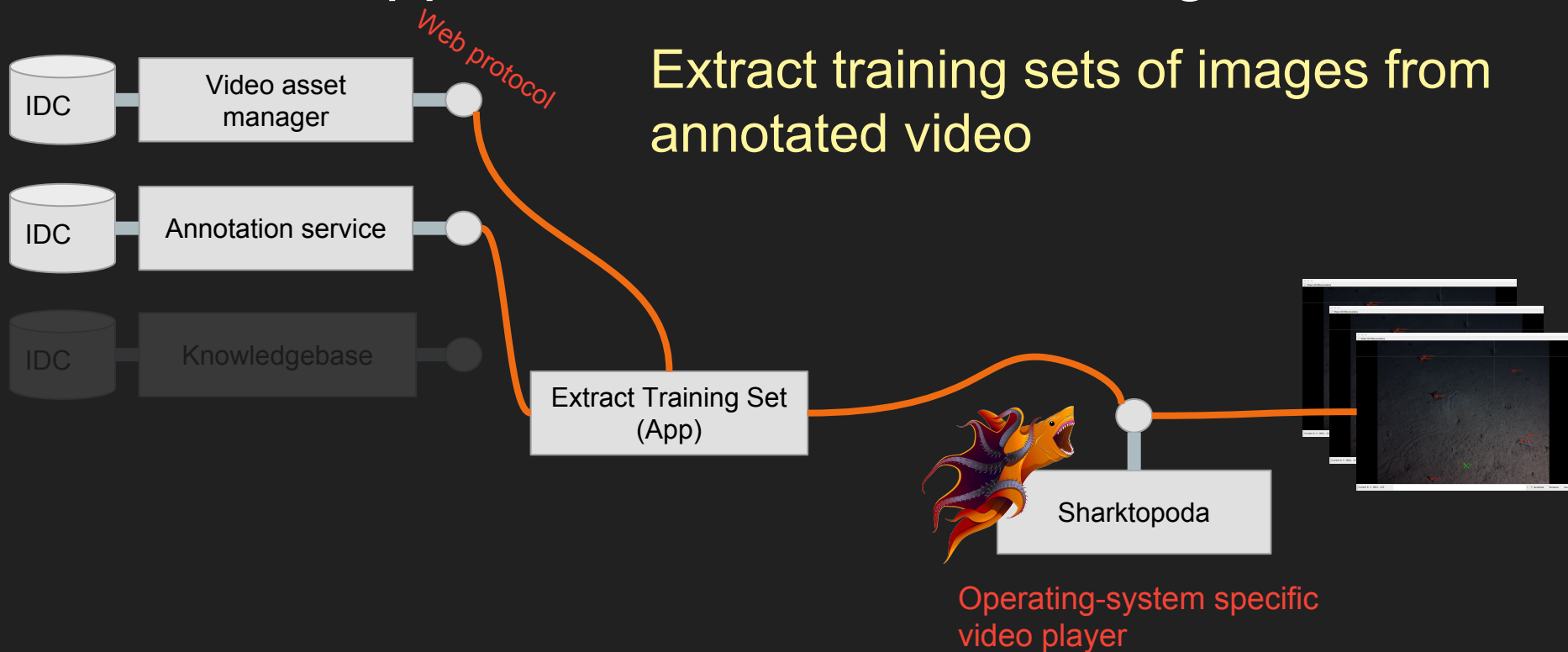
Sharktopoda

video player

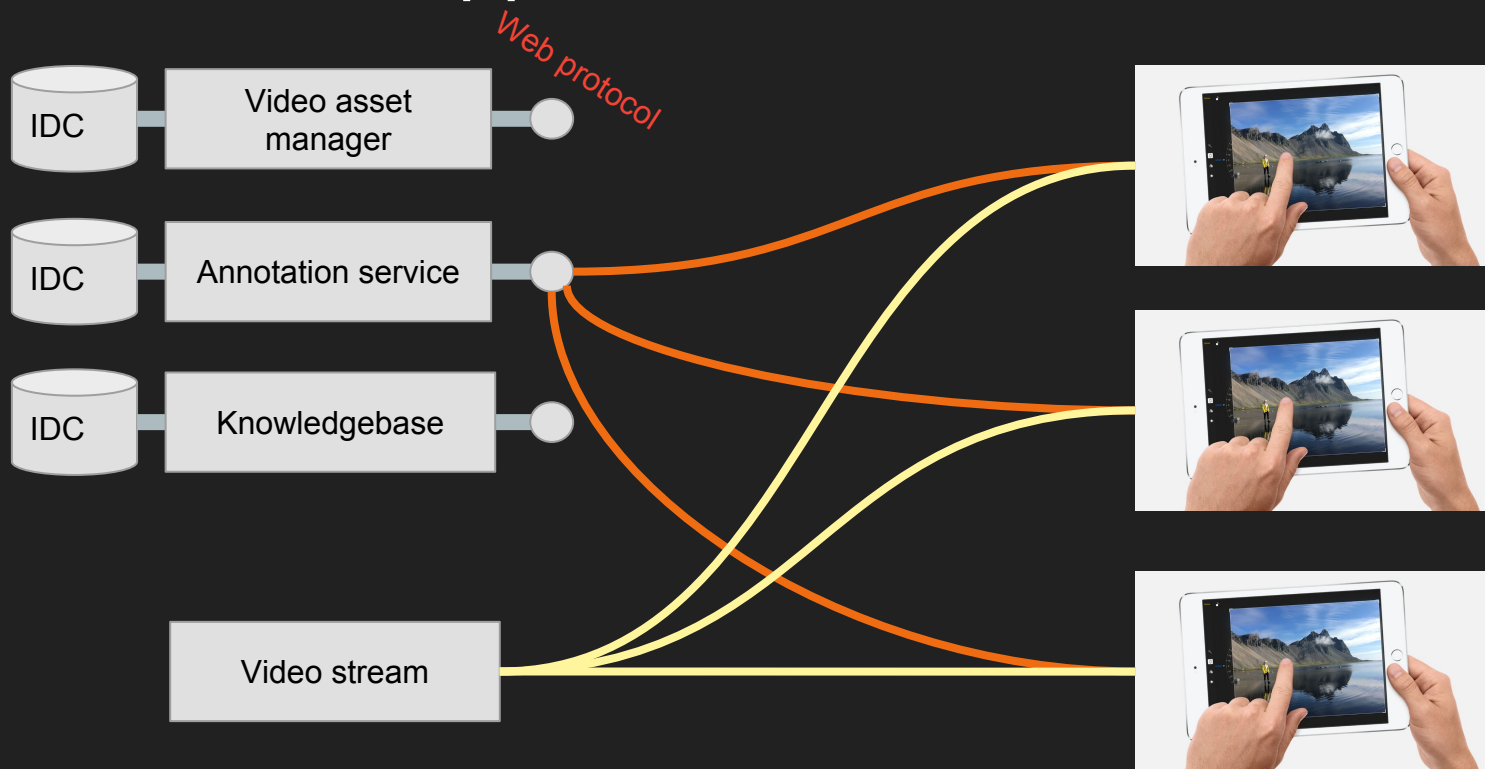
ic

Potential Applications - Extract training sets

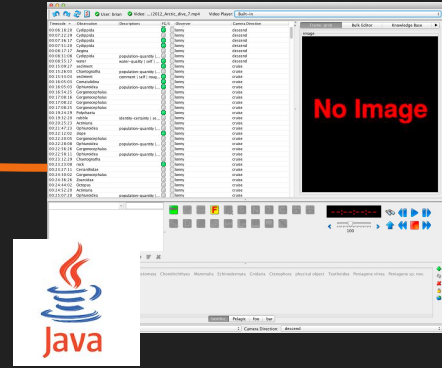
Extract training sets of images from annotated video



Potential Applications - Real-time multi-users



The diagram illustrates the architecture of the Video Asset Manager. On the left, three vertical components are shown: 'IDC' (represented by a cylinder icon), 'Video asset manager' (a rectangle), and 'IDC' (cylinder), 'Annotation service' (rectangle), and 'IDC' (cylinder), 'Knowledgebase' (rectangle). Each 'IDC' is connected to its corresponding service by a horizontal line. The 'Video asset manager' and 'Knowledgebase' have small white circles at their right ends. A red curved line labeled 'Protocol' connects the 'Video asset manager' to the 'Automated Detector and Classifier'. A straight orange line connects the 'Annotation service' to the 'Automated Detector and Classifier'. A red curved line connects the 'Knowledgebase' to the 'Automated Detector and Classifier'.



Develop special tools while using existing applications.

Annosaurus Tutorial

This python3 notebook demonstrates the usage of the [Annosaurus](#) API which is used for creating and editing video annotations. To get started you will need to start annosaurus. If you have [Docker](#) installed you can spin up annosaurus for testing with:

```
docker run --name=anno -p 8080:8080 hohonuuli/annosaurus
```

If you do not have [Jupyter notebook](#) installed you can launch it with:

```
docker run -i -t --name=jupyter \  
-p 8888:8888 \  
-v /Path/To/Dir/With/This/Notebook:/opt/notebooks \  
hohonuuli/jupyter /bin/bash \  
-c "/opt/conda/bin/jupyter notebook --notebook-dir=/opt/notebooks --ip='*' --port=8888 --no-browser"
```

Annosaurus Overview

Annosaurus is a simple [REST/JSON](#) service. When you send an HTTP request, it will respond with JSON content (usually). The JSON response is very simple to parse in most modern programming languages.

In the sections below, the various REST calls and there responses are demonstrated.

Define URLs

Important: In order to run this script you will need to provide your computers inetaddress or computer name as the andpoint.

Normally in an app or script you just define the endpoint and build the other API urls from that.

```
In [1]: # Define endpoint.  
endpoint = "http://10.0.1.251:8080"  
  
annotation_url = "%s/v1/annotations" % (endpoint)  
image_url = "%s/v1/images" % (endpoint)  
image_reference_url = "%s/v1/imagereferences" % (endpoint)  
imaged_moment_url = "%s/v1/imagedmoments" % (endpoint)  
observation_url = "%s/v1/observations" % (endpoint)  
association_url = "%s/v1/associations" % (endpoint)
```

github.com/underwatervideo

underwatervideo.slack.com

the David &
Lucile Packard
FOUNDATION



The screenshot shows the GitHub organization page for "underwatervideo". The browser tabs at the top include "Next-generation Annotation M...", "Home", "Annosaurus Tutorial", "underwatervideo", and "Brian". The address bar shows the URL "https://github.com/underwatervideo". The organization's profile picture is a red 3D model of a submarine. The page header includes navigation links: "This organization", "Search", "Pull requests", "Issues", and "Gist". Below the header, there are tabs for "Repositories", "People 10", "Teams 0", "Projects 0", and "Settings". The "Pinned repositories" section features a card for "UnderwaterVideoWorkingGroup" with the description "Living documents containing recommendation for the acquisition, storage, cataloging and dissemination of underwater video." and 1 star and 5 forks. A search bar and filters for "Type: All" and "Language: All" are present, along with a "New" button. The repository list includes "annosaurus" (Service for storing and retrieving video/image annotations, updated 2 days ago), "vars-kb-server" (Forked from hohonuuli/vars-kb-server, updated 5 days ago), and "vampire-squid" (A video asset manager for tracking videos and camera deployments, updated 7 days ago). A sidebar on the right shows "Top languages" (Scala, C++, Java, Swift, Jupyter Notebook) and "People" (10 members).

The end