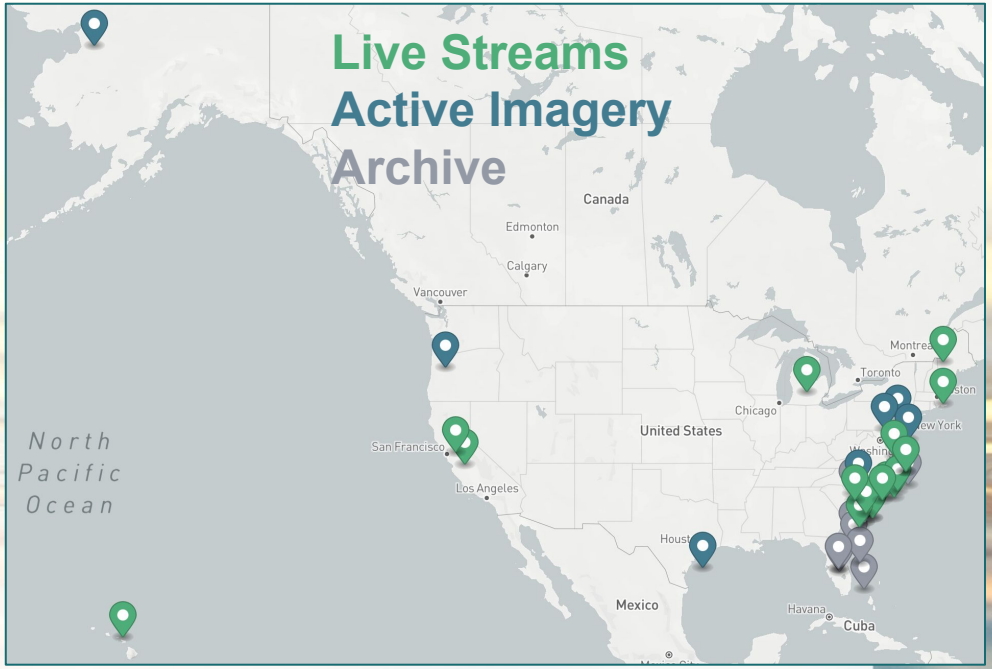
IRA Topic 2: Web Cameras - Data & Imagery Management

Theo Jass (SECOORA), Josh Rhoades (Axiom Data Science)

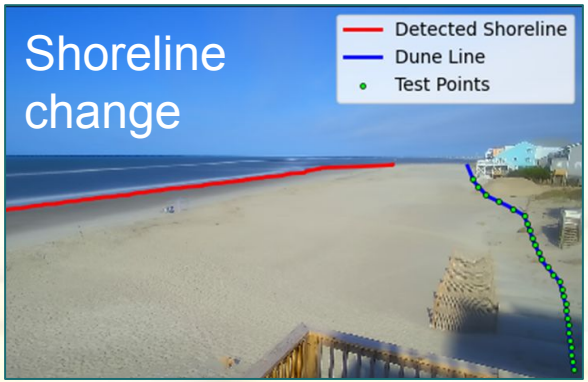
IOOS Annual DMAC Meeting | 1 May 2025

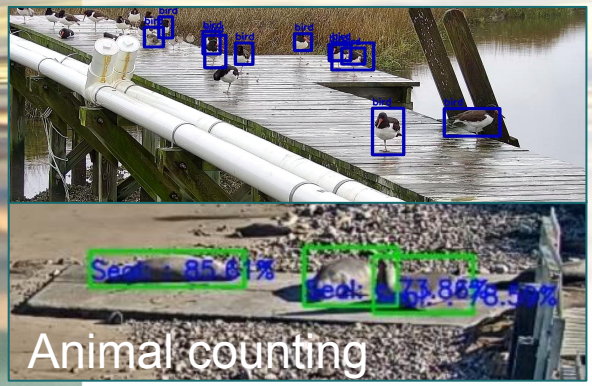
WebCOOS: Webcam **Co**astal **O**bservation **S**ystem

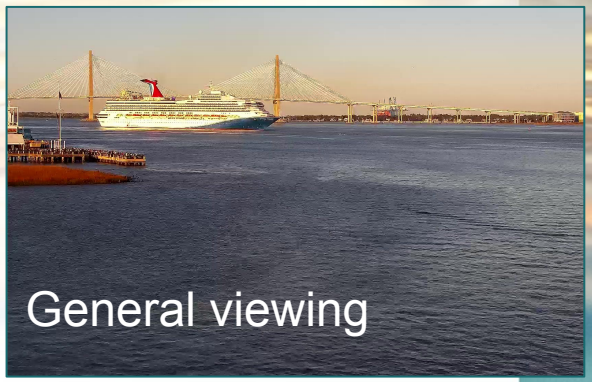
- WebCAT
  - 8 cameras in SECOORA region
- Launching WebCOOS
  - ~20 cameras, mainly SECOORA
  - Applications: AI ML, shoreline change, flooding, beach usage, rip currents
- National WebCOOS
  - +60 cameras across IOOS regions
  - Integrate current algorithms in new places



WebCOOS: Applications





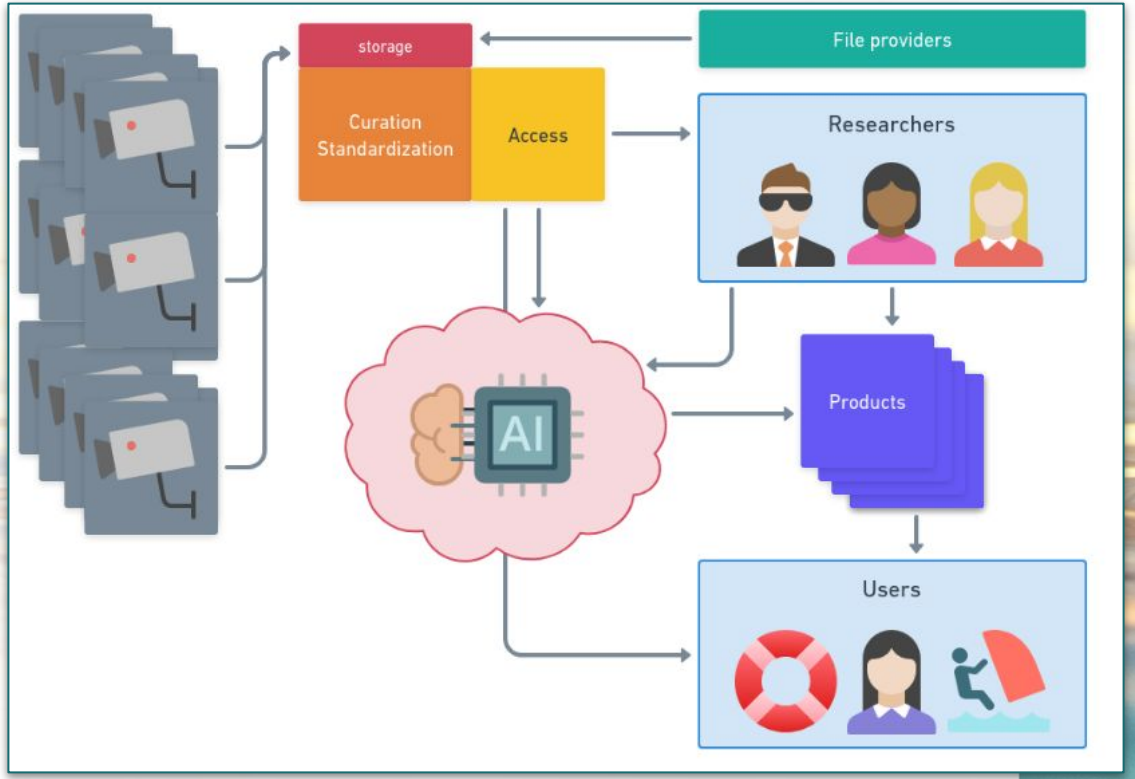


Data management overview: WebCOOS

- Images captured at camera & transmitted to Axiom (via Surfline for new cams)
  - Live stream + 10-minute video chunks + Still images
  - Frequency varies by station & application:
    - From live streaming 24/7 at 30 fps…
    - …to still images every 30 min during daylight hours
- Axiom applies algorithms & stores outputs on Annotated-imagery tab
- Grafana alerts for Axiom + manual checks
- API: webcoos.org/docs & applications
- Data sharing permissions
  - Imagery freely available online
  - Bulk download files >90 days old = prefer wget method with on-site storage

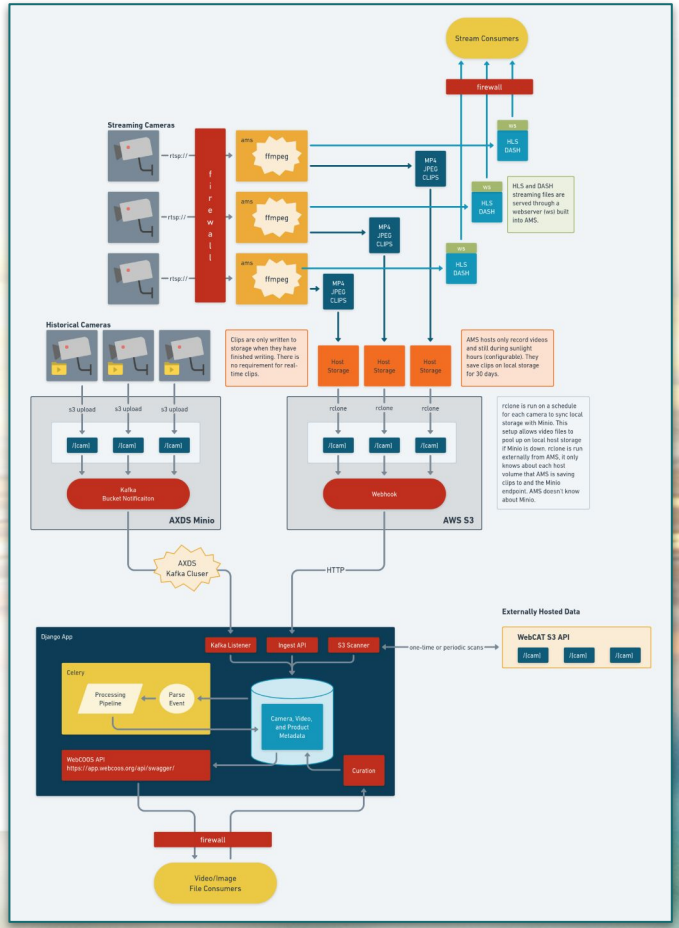
Overall data management pipeline and workflow

- Real-time video stream ingests over the network, or externally provided media (S3 upload)
- Post-processing for:
  - Media metadata indexing, and…
  - Product generation (detections, time averages, brightest pixel)
- Website & HTTP/REST API access for results



Overall data management pipeline and workflow

- Includes “live” streaming products (HLS/DASH streams over HTTP)
- Multiple S3 storage options
- Ingest orchestration via Ansible and Docker
- Live stream & downstream product monitoring via Prometheus/Grafana
- Post-processing via custom Python scripts & web services wrapping researchers’ code provided to Axiom.

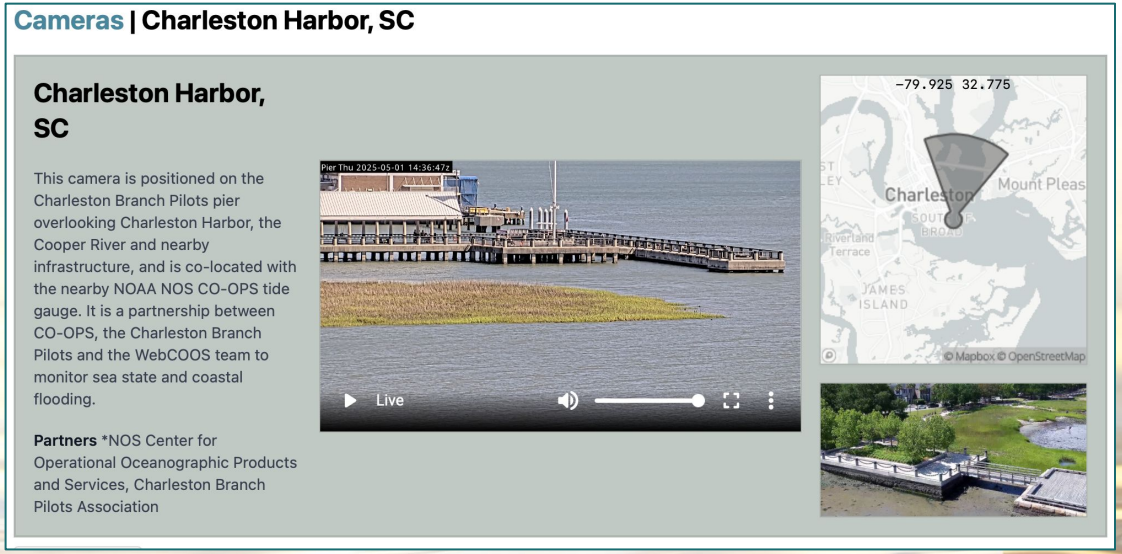


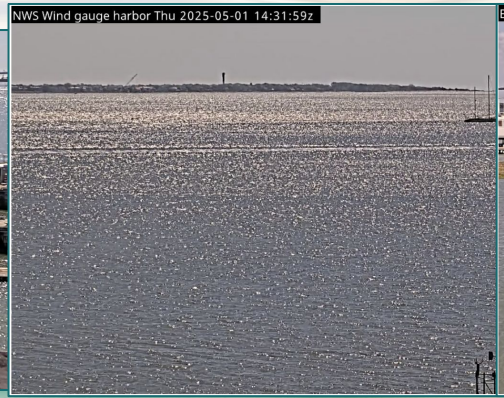
Integrating new cameras (standardized vs. COO)

- New, standardized installs (via Surfline):
  - Standard camera hardware & options → streamlined ingests, fewer unexpected behaviors
  - (Hopefully) easier hardware replacements or upgrades over time
  - Better control over hardware
- Cameras of Opportunity (COO):
  - Network, power, & access methods may exist, but have to be discovered (and likely debugged)
  - Tend to be labor intensive, less reliable, and can exhibit unexpected behaviors.
  - Can’t often be remotely managed.

General camera metadata

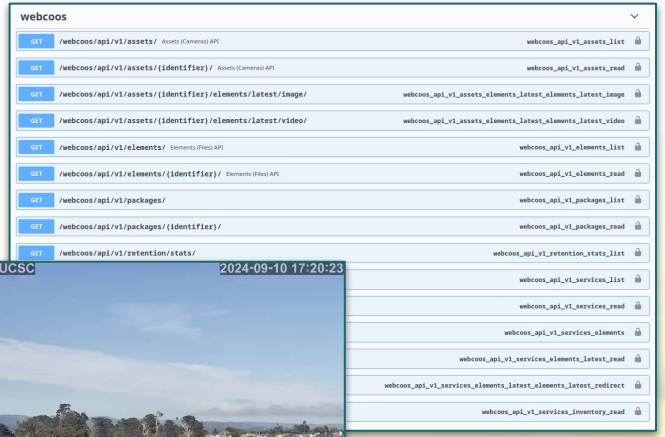
- Direction
- Location
- Start/End date
- Status
- Machine & human friendly naming
- Description
- Partners
- View?

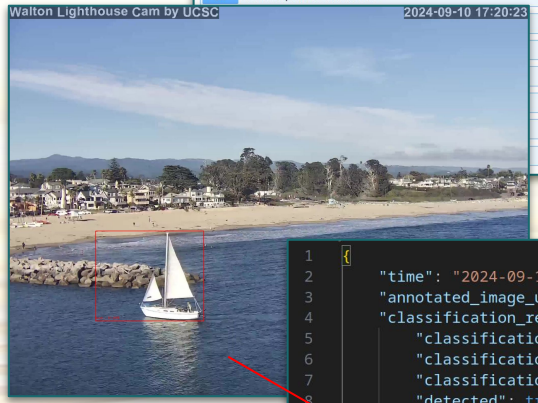




Camera metadata, continued

- Media files with metadata indexed in Postgres (available via WebCOOS API request)
  - Time extent (duration)
  - Size
  - Location (lat/lon)
- JPG and MP4 files updated with EXIF metadata
  - Lat/Lon, camera title, UTC timestamp in basename
- JSON files further annotate ML detection & shoreline detection results






```
{
    "time": "2024-09-11T00:20:57Z",
    "annotated_image_url": "https://stage-webcoos-object-c
    "classification_result": {
        "classification_model_framework": "sahi",
        "classification_model_name": "yolo",
        "classification_model_version": "v8n",
        "detected": true,
        "detection_count": 1,
        "classification_scores": [
            {
                "boat": 0.9108595252037048
            }
        ],
        "classification_bboxes": [
            [
                {
                    "x": 427,
                    "y": 1107
                },
                {
                    "x": 960,
                    "y": 1550
```


OpenAPI spec: https://app.webcoos.org/api/swagger/

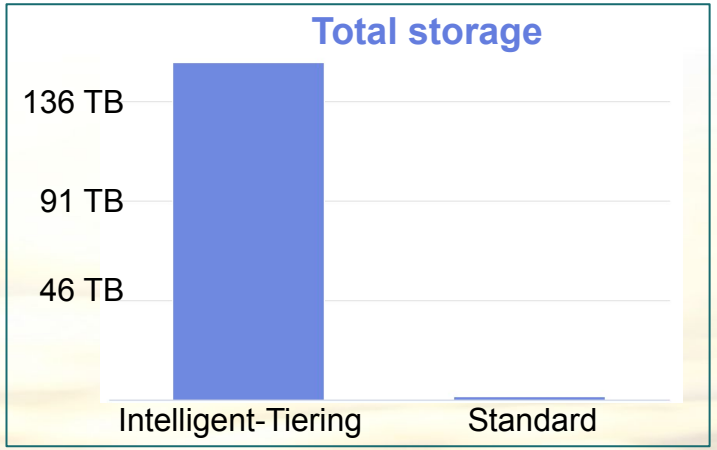
Integrating code from existing algorithms into new cameras

- Existing:
  - ML detections (Rip current models, Seal model, General-purpose object detection model)
  - Shoreline Detection (with brightest pixel & time averaged pixel products)
- Uses Axiom compute resources (CPU, GPU, storage, network)
- Lots of compute & labor intensive:
  - Code collaboration
  - Adapting existing code to HTTP services
  - Testing & performance profiling
- Using algorithms at new sites
  - Rips - site-specific challenges (different angles, types of rips, etc)
  - Shoreline - site-specific config files
  - Objects - use YOLOv8 & COCO datasets (person, umbrella, etc.)

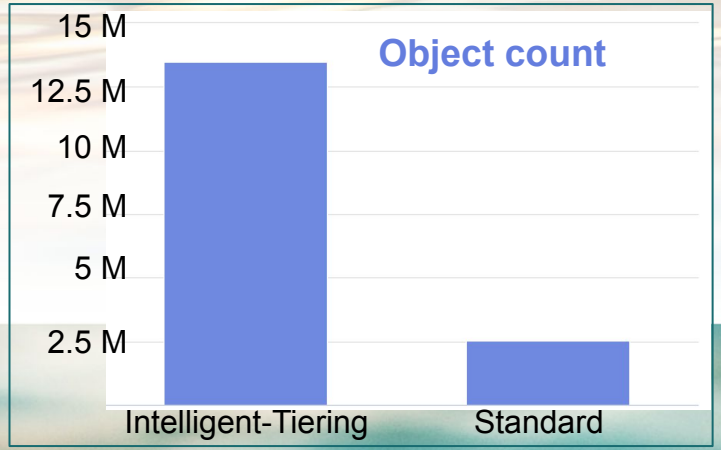
5 minute load average over 48 hours (9/25/2024)

Technical requirements for different product applications

- CPU and GPU compute
  - ML model detection tends to require GPU support (both GPU compute and VRAM)
- Scaling: number of cameras over time
  - ~200GB a day
  - ~300-700 media files per asset per day (potentially need processing)
- Currently: 9 physical hosts, 3 GPUs

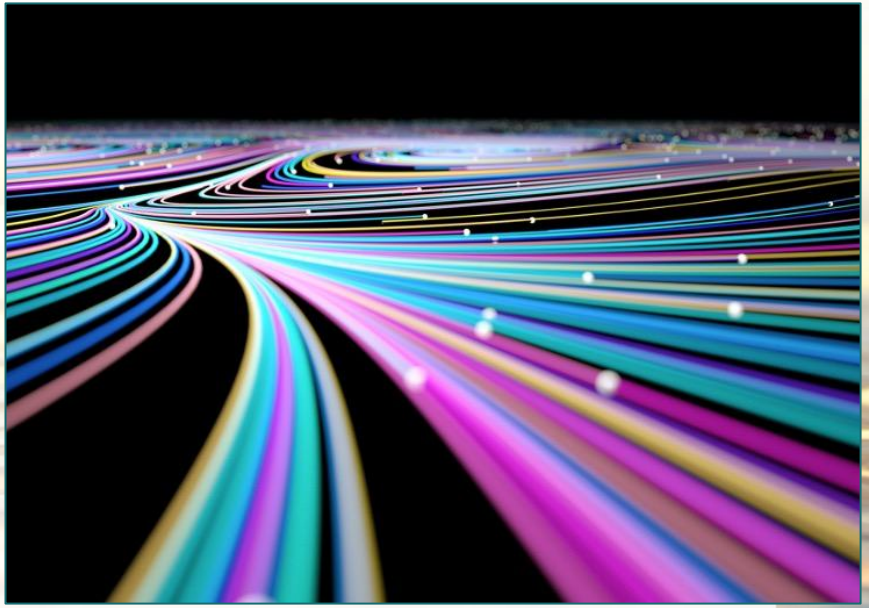


156TB among **16 million** media files



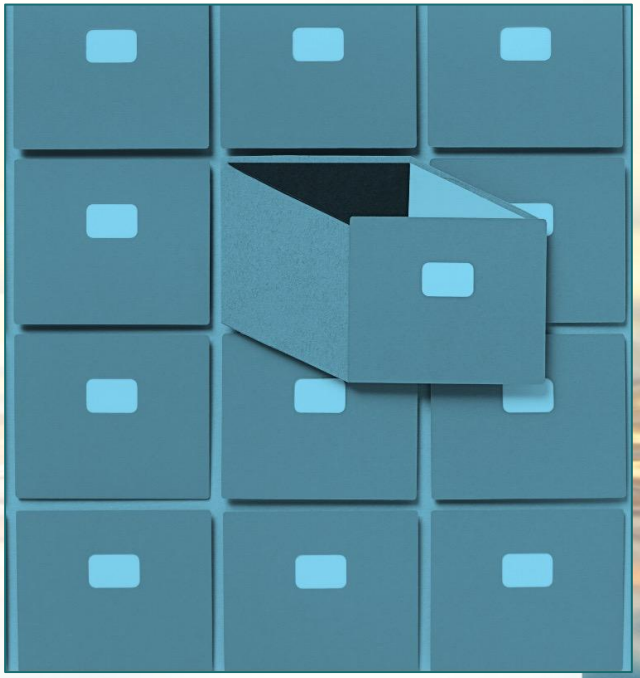
Data access methods

- Incoming (ingress to AXDS):
  - RTSP/RTMP (“push” or “pull”, depending on network situation)
    - Similar to Youtube, Twitch, etc.
  - HLS/DASH streams (via HTTP)
  - S3 upload
- Outgoing (egress from AXDS):
  - webcoos.org (website)
  - WebCOOS API (HTTP/REST)
  - HLS/DASH streams (via HTTP)
  - S3 download
  - HTTP indexed directories (less used)



Data archiving: WebCOOS

- Standard = keep everything on webcoos.org & API
- AWS S3 Intelligent Tiering
  - Different tiers of storage, based on last access date
  - Older & “colder” = cheaper
  - If an old/cold file is accessed, its timeline resets
- Working to implement retainment policies to save data storage costs, e.g.:
  - NWLON stations = delete videos from webcoos.org after 30 days if water level did not reach NOS-Minor flood levels
- WebCAT (FY17-21): archived on webcoos.org but not readily accessible (thumbnails only)
- **Discussion**: how long to keep different kinds of files?

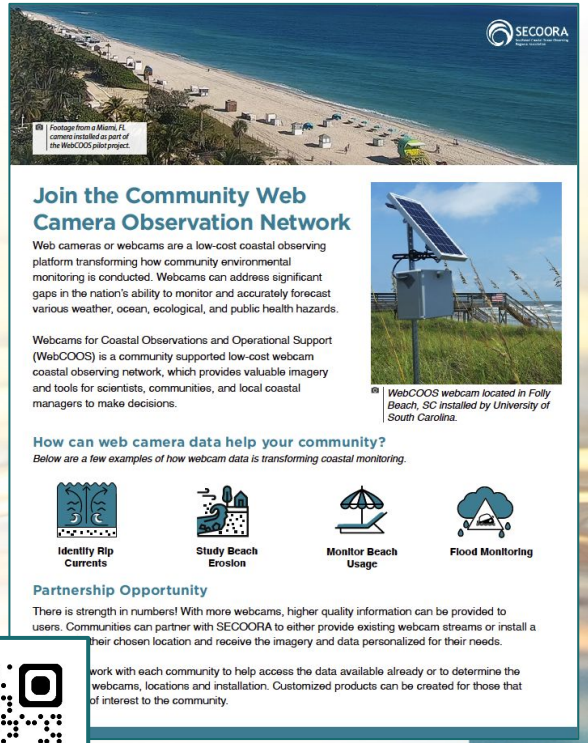


How are we getting data to NOAA/other federal programs (or where do we want it to go)?

- NOAA
  - Coastal Inundation Dashboard
  - NWS: Marine Warnings (small craft)
- USGS: CoastCams
- USACE: ERDC-CHL CorpsCams
- Validate models/forecasts:
  - NOAA & IOOS RA modelers
  - Total Water Level and Coastal Change Forecast
  - Rip currents, with NWS?
- Ecological monitoring
- Data.gov? (longer-term storage/access)
- **Discussion:** Are there more or better ways to do this?
  - Monthly/Annual High Tide Flooding Products?
  - CORA/CoSMoS/HERA?
  - HIVIS?
  - USCG? SAR?
  - Ecological monitoring - NCCOS?

How are we getting data to RA end users/ community members/data products?

- webcoos.org + “Annotated-images” tab
- API & saludasys (counts, alerts, etc)
- Local governments/managers
- Local lifeguards/beach safety
- Community members & groups (partners)
- Camera hosts
- Local WFO
- Local news/weather outlets/Weather Channel
- Nonprofit & research partners, including private sector/consultants
- Small businesses
- Beach Conditions Reporting System, ShellCast, other partners
- Outreach & social media, including event-based
- Pilots/maritime domain awareness
- Survey out now on website updates & improvements:
- NWLON WebCOOS Synchronizer- **stay tuned!**



DMAC goals for IRA Topic 2

- Continue data storage & management from OTT cameras
  - Physical O&M not covered
- Provide robust national DAC as cameras from OTT & IRA are added, streaming, and storing data
  - 20 → 80 → 100+ cameras
- Add some new web cameras (beyond OTT) & store/manage their data
  - Average of ~3 per RA, plus any extras contracted by individual RAs
- Continue to manage, update, and upgrade webcoos.org
- Integrate some new features/algorithms as they are desired by end users, developed by researchers, and capacity allows

Questions?

- Open questions within WebCOOS for the breakout or for others:
  - Data storage
    - How long to keep different kinds of files?
  - Ways to get data to partners
    - More or better ways to get data to federal partners?
  - What are others currently doing with camera data?
  - What are our data goals & objectives for IRA Topic 2 web camera data?
  - How should we continue to meet regularly to make collective progress?
- Primary WebCOOS contacts at Axiom:
  - Josh Rhoades, Software Engineer (josh@axds.co)
  - Dr. Karina Khazmutdinova, Project Manager (karina@axds.co)
  - James Doyle, Project Manager (james.doyle@tetratech.com)

Sunset at Miami Beach

Technical requirements for different product applications

- Rip Current Detection
  - Requires NVIDIA GPU acceleration (40+ CPU cores cannot serve as a replacement)
  - Support multiple rip current models at once
  - Multi-stage products
    - input video → 15-second time average → ML model (general) → detection output
    - input video → once-a-minute still → ML model (walton) → detection output
  - Models as developed work with real-time video streams, but Axiom's pipeline isn't wired for that
  - 2 models developed, 1 deployed (visual appearance)- Accuracy depends on strength of visual signature of rip
- Shoreline
  - **[Filtered]** Multi-stage products:
    - [Input video, > 9 minutes, top of the hour] → ~10 minute time average → shoreline detection algorithm → shoreline detection product
  - GPU provides negligible improvement, high CPU during time average and brightest pixel calculations.
  - Heavy config lift for new cameras
- General Object Detection
  - Works well, have to tweak detection:
    - Filtering out erroneous object detections (trains on beach, etc.)
    - Use a "windowing" feature to better identify objects in frame.
  - PTZ: different object detection results vs fixed camera view