

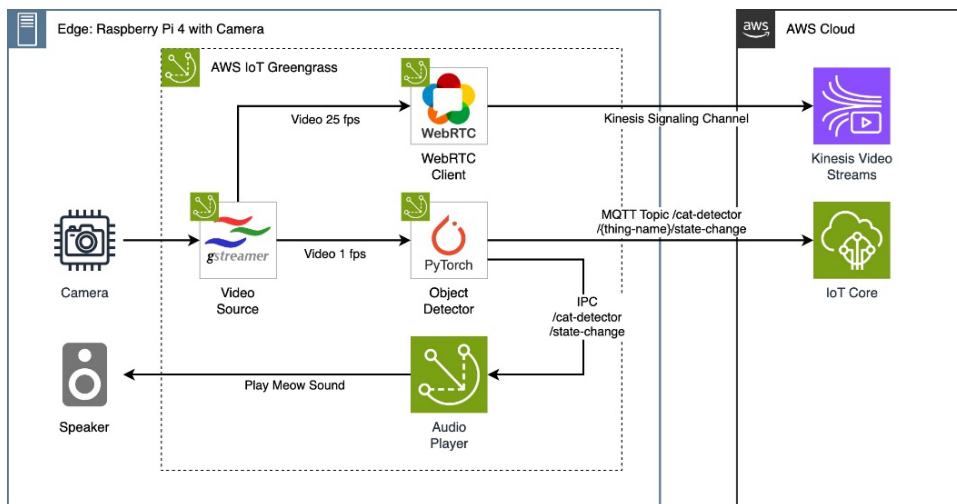
# Amazon Kinesis Video Streams Cat Detector Demo Product Design

Today, an astounding 80% of global web traffic comprises videos, yet a significant portion of these videos go unseen by human eyes. Instead, they are increasingly scrutinized and processed by artificial intelligence (AI) systems, much like the cat detector demo showcased here.

Amazon Kinesis Video Streams (KVS) is a cutting-edge solution that enables secure video ingestion into the cloud, offering features such as indexing, video recognition, analysis, and machine learning capabilities. With KVS, the creation of real-time video-enabled applications supporting live, recorded, and two-way streaming is achievable at scale.

## KINESIS VIDEO CAT DETECTOR OVERVIEW

The Cat Detector is an IoT application that uses machine learning to detect cats using an trained ML model running on a low-cost Raspberry Pi. Potential use cases include keeping your cats off the kitchen counter or away from houseplants. The application consists of two components - an IoT application running on the Pi, and a website hosted in the cloud.

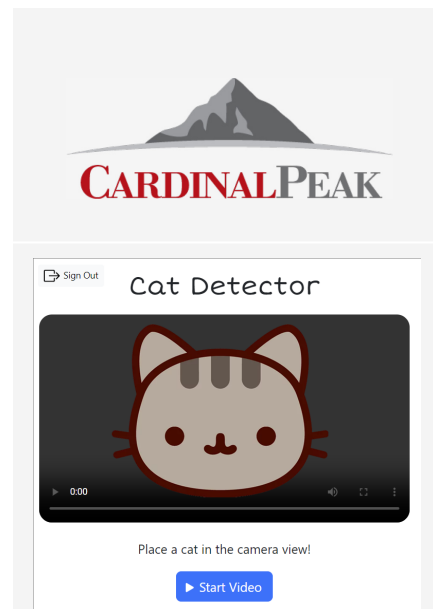


## CAT DETECTOR RASPBERRY Pi IoT APPLICATION

The Cat Detector is an [AWS IoT Greengrass](#) application that runs continuously on the Pi as a set of background processes. It has the following components:

- **AWS IoT Greengrass Nucleus** - The main controller for the other components is [AWS IoT Greengrass Nucleus](#). This is a standard AWS IoT Greengrass component that handles starting and stopping the other components, configuration, logging, component software updates, etc.
- **Video Source** - This component streams video from the Pi camera using a GStreamer pipeline, and shares it with the other components via two separate copies of the original stream, each having different formats:
  1. The first stream sends 640 x 360 resolution 25 frames per second I420 BT.601 raw video to the WebRTC Client component.
  2. The second stream sends 640 x 360 resolution one frame per second RGB 1:1:16:4 raw video to the Object Detector component.

This component is a bash script that uses the `gst-launch-1.0` command to create the GStreamer pipeline. It uses hardware acceleration to reduce the load on the Pi CPU.



## OUTSOURCED ENGINEERING

Cardinal Peak harnesses the power of AWS services like KVS to collaborate with clients in conceiving, launching and supporting groundbreaking digital and physical products. As a recognized [AWS IoT Service Delivery partner](#), we specialize in KVS, streaming video technologies, and AWS IoT product engineering services.

## CLIENT VALUE

- Bring products to market quicker
- Lower development risk
- Offload your internal management
- Add the right experience to your team
- Get CTO level input



## CARDINAL PEAK, LLC

1380 Forest Park Circle  
Suite 202  
Lafayette, CO 80026-3378  
303.665.3962  
[cardinalpeak.com](http://cardinalpeak.com)

## Amazon Kinesis Cat Detector Raspberry Pi IoT Application

- **WebRTC Client** - The WebRTC Client component accepts WebRTC calls from the Website video player via the [AWS Kinesis video service](#). When a call is started, it starts its own GStreamer pipeline using the Video Source as its source, and converts the video into the required H.264 format using hardware acceleration.

The component is a C application, based on an example application from the [AWS Kinesis WebRTC C SDK](#). When a call ends, the component releases its resources so that it is ready to accept the next call.

- **Object Detector** - This component is responsible for detecting cats in the video. It is a Python application that reads RGB video frames from the second Video Source stream once per second. Frames are classified using a machine learning model, and if a cat is detected, a message is published to an [IoT MQTT topic](#) and to an internal [Greengrass IPC topic](#). These messages are picked up by the Website and Audio Player component respectively.

The machine learning model used is a pre-trained ResNet50 model running on the [PyTorch Torchvision](#) library. Each classification takes approximately 2.5 seconds to run on the Pi, and only one classification runs at a time. This means that one in every three video frames is classified, and there is a delay of approximately five seconds between the cat being put in the Pi camera view, and the publishing of the detection message.

- **Audio Player** - The Audio Player component is responsible for playing a short audio clip each time a cat is detected. It subscribes to a local Greengrass IPC topic, and when a cat detection message is received, it plays a WAV file through a speaker plugged into the Pi headphones jack.

## KINESIS CAT DETECTOR CLOUD COMPONENTS

- **Website** - The website is a static web application written in React. It uses the [AWS Amplify SDK](#) and [AWS Kinesis WebRTC SDK](#) libraries and is hosted via Github Pages. It has four main features:
  1. A video player that allows users to view the live video from the Raspberry Pi camera. It works by starting a WebRTC call with the WebRTC Client component on the Pi, via the AWS Kinesis Video service.
  2. When a cat is detected, a cat image pops up above the video player. This function works as follows: when the page loads, it connects to AWS IoT and subscribes to an MQTT topic. When the Object Detector component on the Pi detects a cat, it sends a message on this topic that is received by the web page.
  3. Users are authenticated using [AWS Cognito](#) and must sign in via a Cognito hosted login page. Once signed in, the Cognito session provides the necessary AWS credentials for the web page to authenticate to the AWS IoT and AWS Kinesis WebRTC services.
  4. An information section displays instructions how to use the demo, technical information about the architecture of the Cat Detector application, and information about AWS IoT Consulting Partner Cardinal Peak.
- **AWS Greengrass** - The initial set-up and configuration of the Pi, and subsequent component software updates are performed manually using a combination of the AWS Console and running local [Greengrass CLI commands](#).
- **AWS Kinesis Video Signaling Channels** - This service is used for the WebRTC video calls, but no manual configuration steps are necessary. The AWS Console can be used for debugging and development.
- **AWS Cognito** - User authentication is handled by AWS Cognito. A User Pool is configured to provide a hosted login page for the application. A corresponding Identity Pool is used to authorize Cognito sessions to access the necessary resources in AWS IoT and AWS Kinesis Video services.
- **IAM Roles and Profiles** - The Cognito Identity Pool has an authenticated access role attached, which has a policy to allow access to IoT and Kinesis Video. Only the minimum necessary permissions are granted.  
The Pi gets the necessary access to Greengrass, IoT and Kinesis Video via a policy attached to its IoT certificate.

## Need Help Bringing Your Project to Life?

[Reach out today](#) to initiate a conversation about your project. Discover more about our [video product design services](#) and [solutions for audio and video streaming](#) including case studies and expert articles.