Specification and Operation of Privacy Models for Data Streams on the Edge

Boris Sedlak, Ilir Murturi, and Schahram Dustdar

dsg | DISTRIBUTED SYSTEMS GROUP

TU WIEN Informatics

Introduction - Problem Statement (1/2)

P1: increasing number of IoT devices streaming sensor data, privacy enforcement happens in resource-rich cloud environments

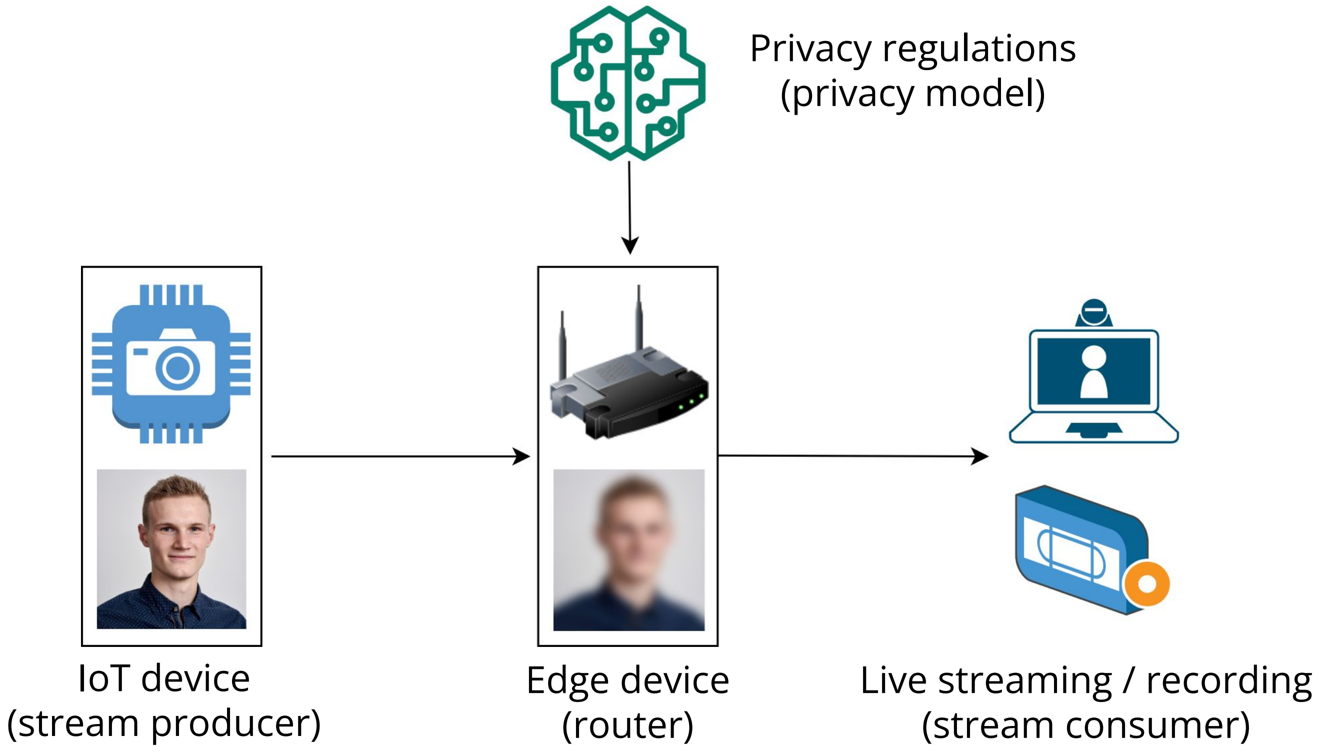
→ high latency and high chance of intercepting data
← processing at powerful edge devices, decrease network traffic

Introduction - Problem Statement (2/2)

P2: increasing number of (written) privacy regulations that must be respected by companies

→ custom implementations for ensuring privacy
← standard description of privacy requirements, smart environment that enforces transformations based on this specifications

Introduction - Solution Attempt

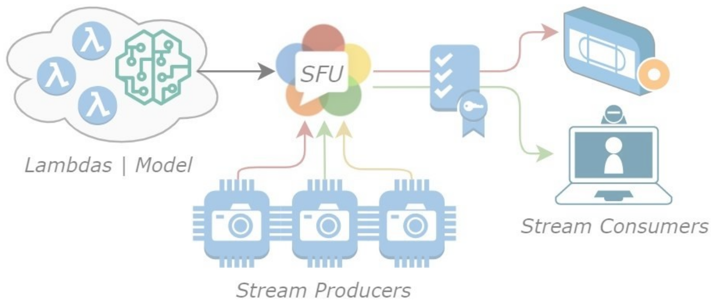


Introduction - IoT vs Edge device

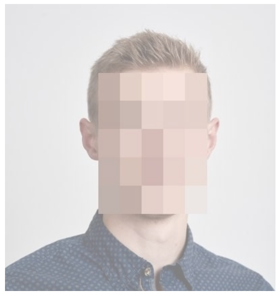
Assume privacy model, enforce directly on **IoT device**?
$\leftarrow$ No common environment, insufficient resources,
restricted update mechanisms
$\rightarrow$ Powerful (interconnected) edge devices (e.g. Nvidia Jetson)

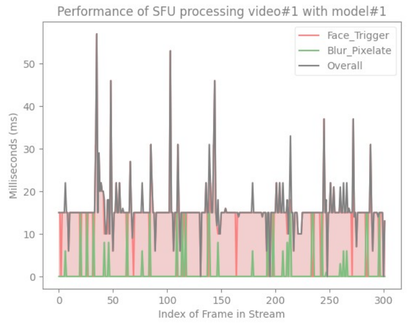
Framework

Abstract Concept



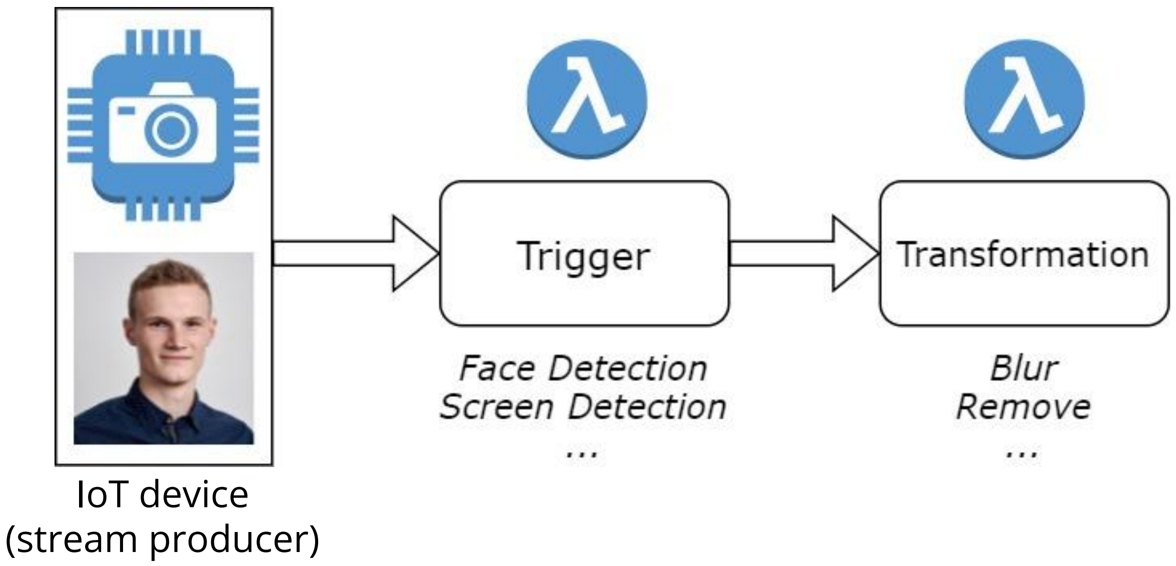
Prototype



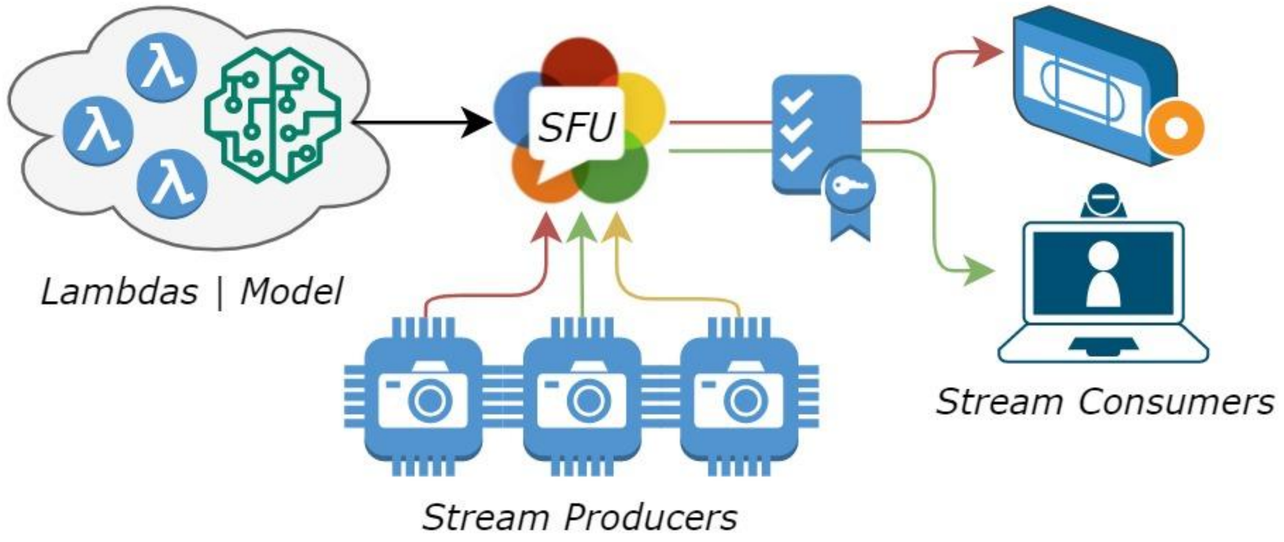


$$Face\_Trigger : \{'prob' : 0.85\} \rightarrow Blur\_Area\_Pixelate : \{'blocks' : 5\}$$

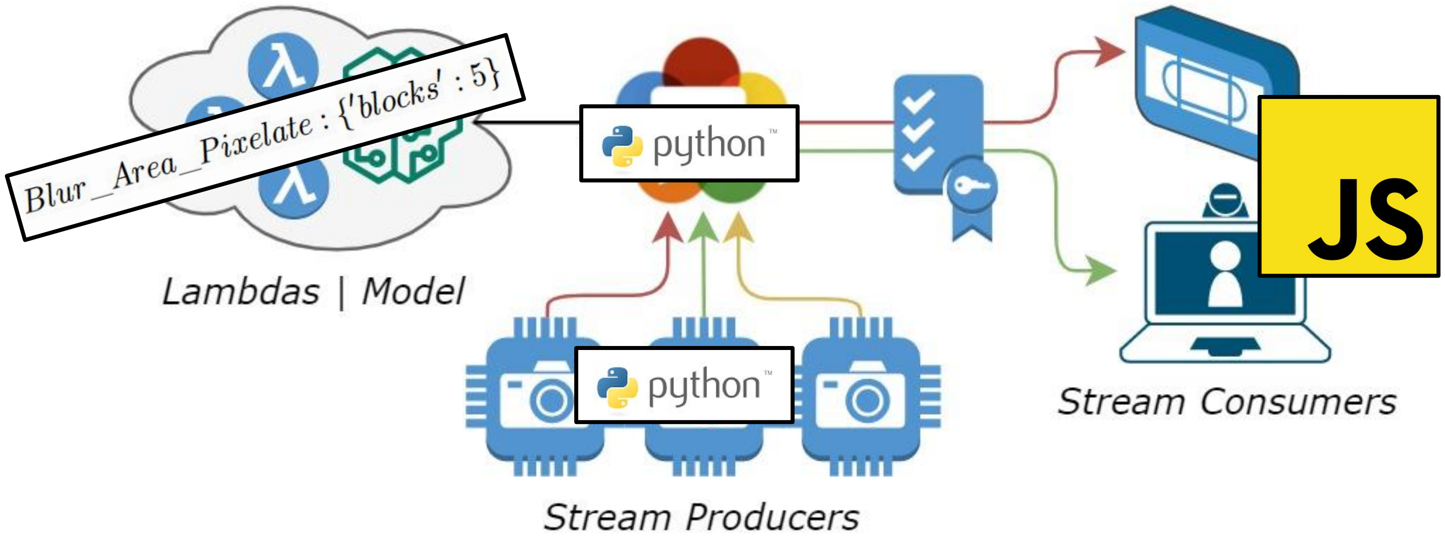
Abstract Concept - Model Specification



Abstract Concept - Architecture

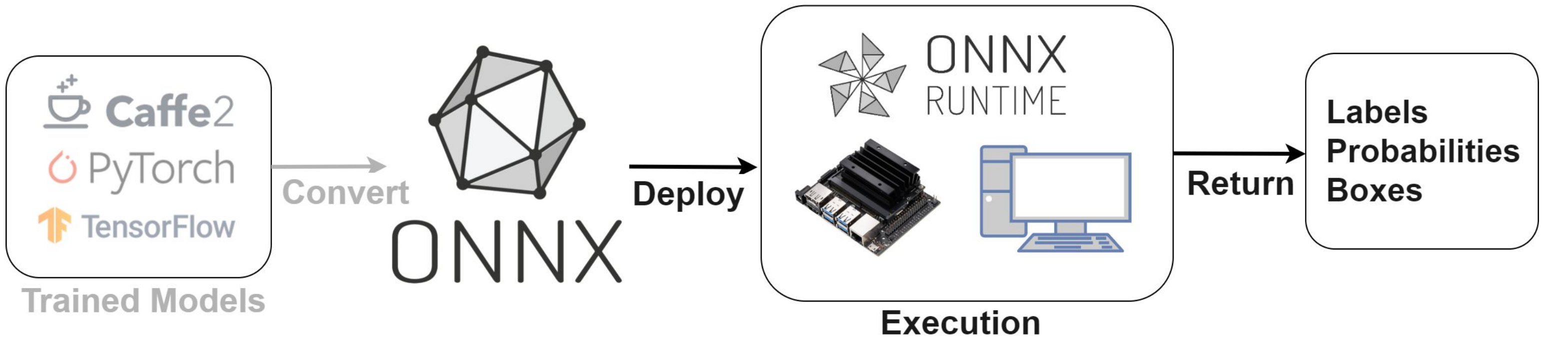


Prototype - Video Streaming

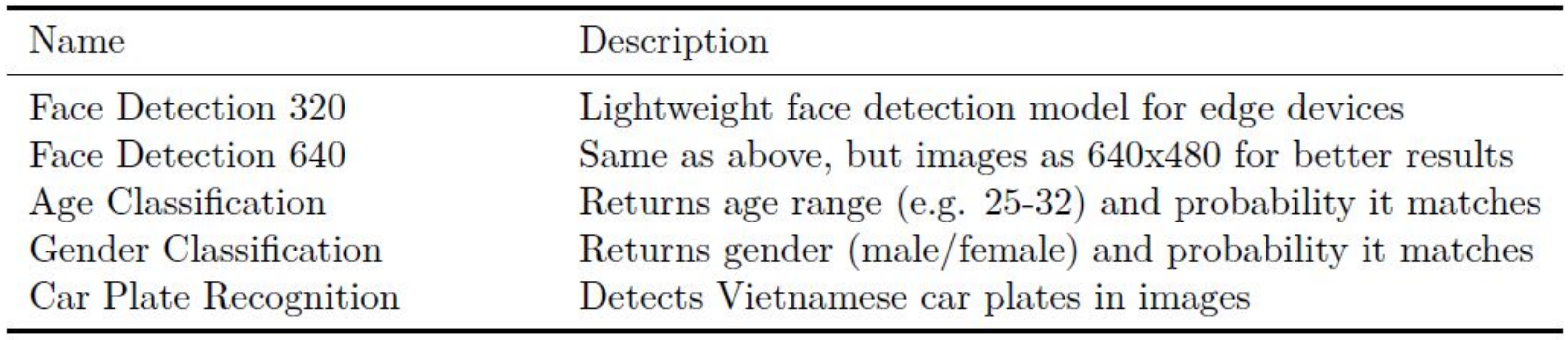


https://en.logodownload.org/python-logo/ https://commons.wikimedia.org/wiki/File:Unofficial_JavaScript_logo_2.svg

Prototype - Pattern Detection (1/2)



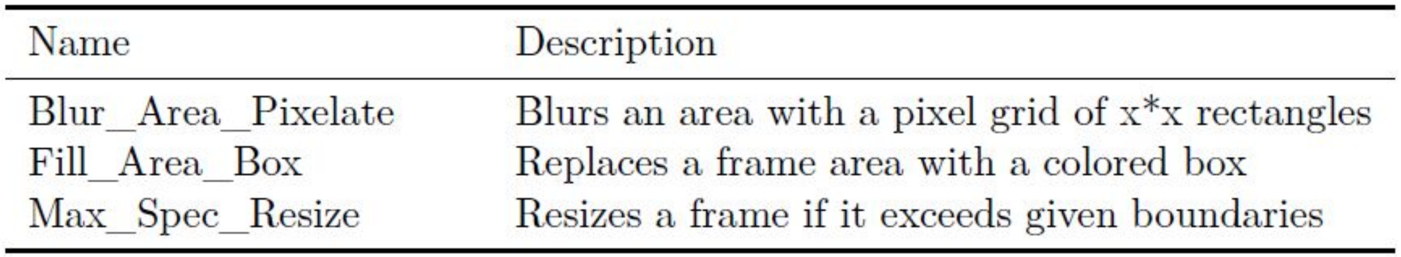
Prototype - Pattern Detection (2/2)



Name	Description
Face Detection 320	Lightweight face detection model for edge devices
Face Detection 640	Same as above, but images as 640x480 for better results
Age Classification	Returns age range (e.g. 25-32) and probability it matches
Gender Classification	Returns gender (male/female) and probability it matches
Car Plate Recognition	Detects Vietnamese car plates in images

ONNX models used for triggers

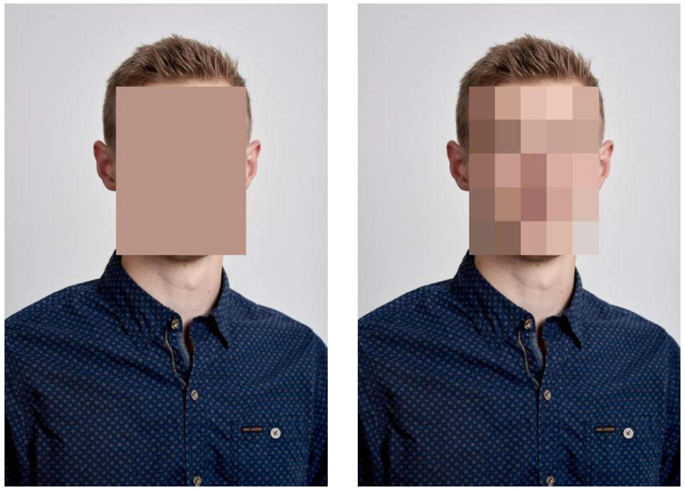
Prototype - Transformation Functions



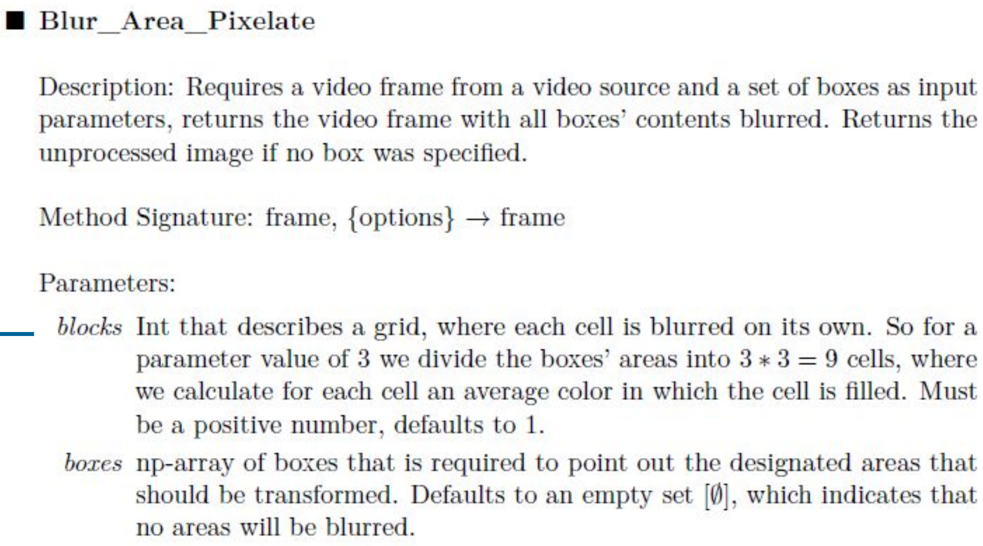
Name	Description
Blur_Area_Pixelate	Blurs an area with a pixel grid of x*x rectangles
Fill_Area_Box	Replaces a frame area with a colored box
Max_Spec_Resize	Resizes a frame if it exceeds given boundaries

Transformation functions

Prototype - Transformation Example



blocks {1,5}



■ **Blur_Area_Pixelate**

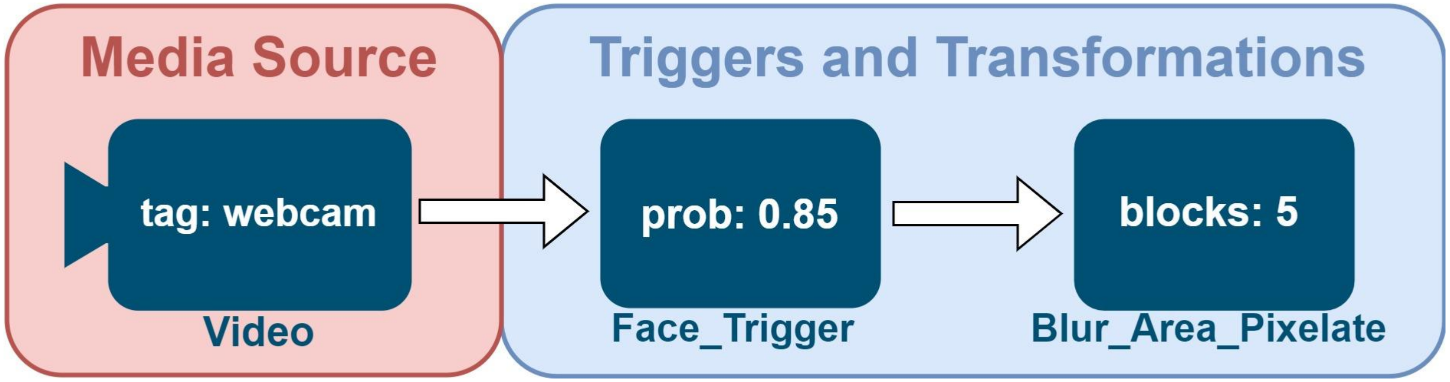
Description: Requires a video frame from a video source and a set of boxes as input parameters, returns the video frame with all boxes' contents blurred. Returns the unprocessed image if no box was specified.

Method Signature: frame, {options} → frame

Parameters:

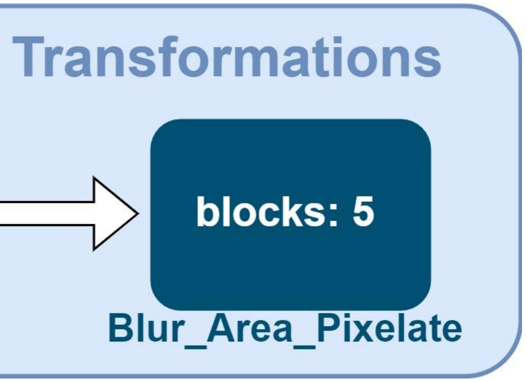
blocks	Int that describes a grid, where each cell is blurred on its own. So for a parameter value of 3 we divide the boxes' areas into $3 * 3 = 9$ cells, where we calculate for each cell an average color in which the cell is filled. Must be a positive number, defaults to 1.
boxes	np-array of boxes that is required to point out the designated areas that should be transformed. Defaults to an empty set [∅], which indicates that no areas will be blurred.

Prototype - Privacy Model / Chain



$$video : \{'tag' :' webcam'\} \rightarrow Face\_Trigger : \{'prob' : 0.85\} \rightarrow Blur\_Area\_Pixelate : \{'blocks' : 5\}$$

Prototype - Privacy Model / Chain



$Blur\_Area\_Pixelate : \{'blocks' : 5\}$

Conclusion

- Specification of privacy requirements
  - Chains of triggers & transformations
- Underlying architecture for policy enforcement
  - Not limited to a specific data type
- Prototype for video streaming
  - Decreased latency by moving resources to edge
  - Accelerated image processing on GPU

Conclusion - Future Work

1. Feature other data types in prototypes
2. Design live monitoring component
3. Evaluate security aspects

Thank you for your attention!
Questions?