TECHNISCHE
UNIVERSITÄT
WIEN

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

Boris Sedlak

Diploma Examination | 26.04.2022

Introduction - Problem Statement

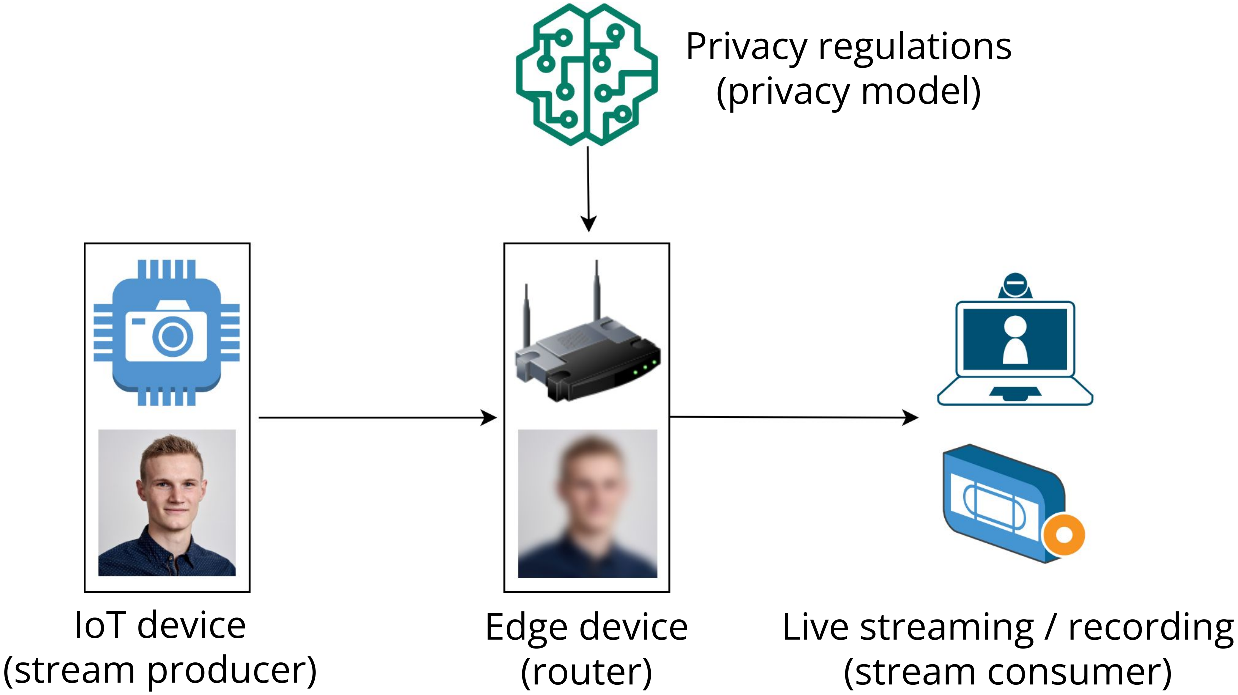
P1: increasing number of IoT devices streaming sensor data, privacy enforcement happens in resource-rich cloud environments

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

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



Privacy regulations
(privacy model)

IoT device
(stream producer)

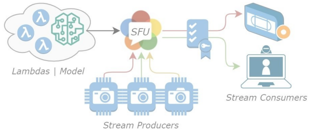
Edge device
(router)

Live streaming / recording
(stream consumer)

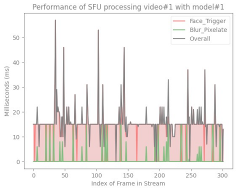
Introduction - Research Questions

1. How can privacy requirements for edge networks be specified and represented as privacy models?

2. What architecture is the most efficient for the distributed execution of privacy models?

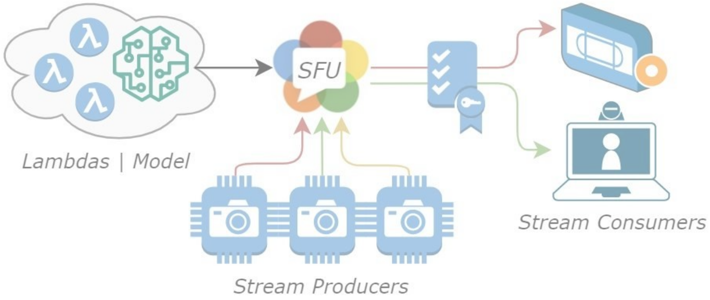


3. Is the presented architecture in fact able to transform a data stream according to a privacy model within a respective time span?

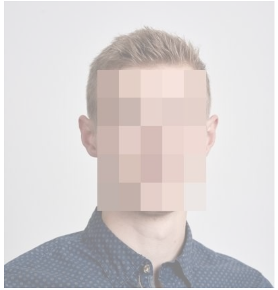


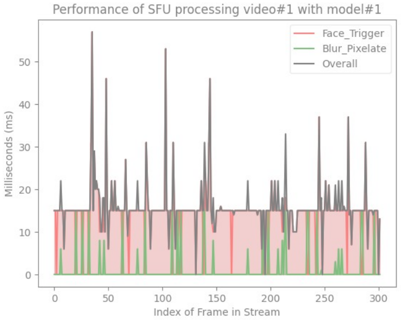
Framework

Abstract Concept



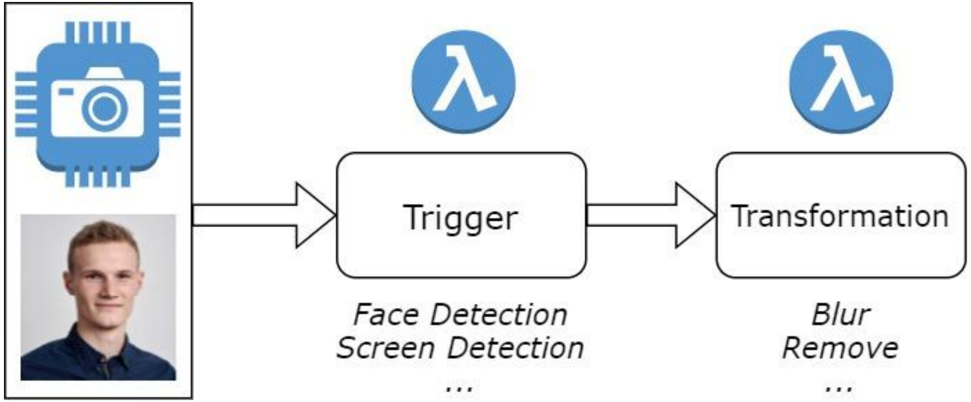
Prototype





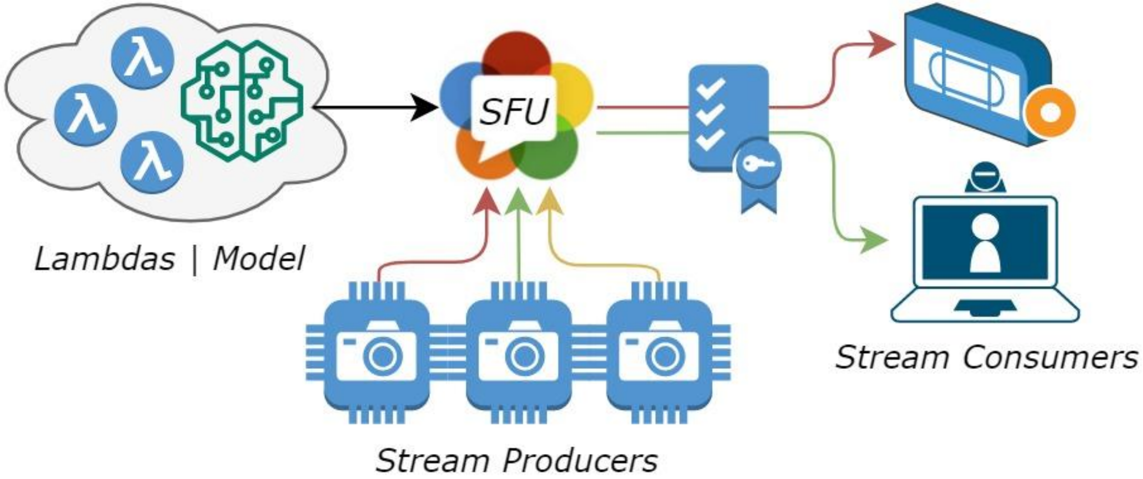
$$Face\_Trigger : \{'prob' : 0.85\} \rightarrow Blur\_Area\_Pixelate : \{'blocks' : 5\}$$

Abstract Concept - Model Specification

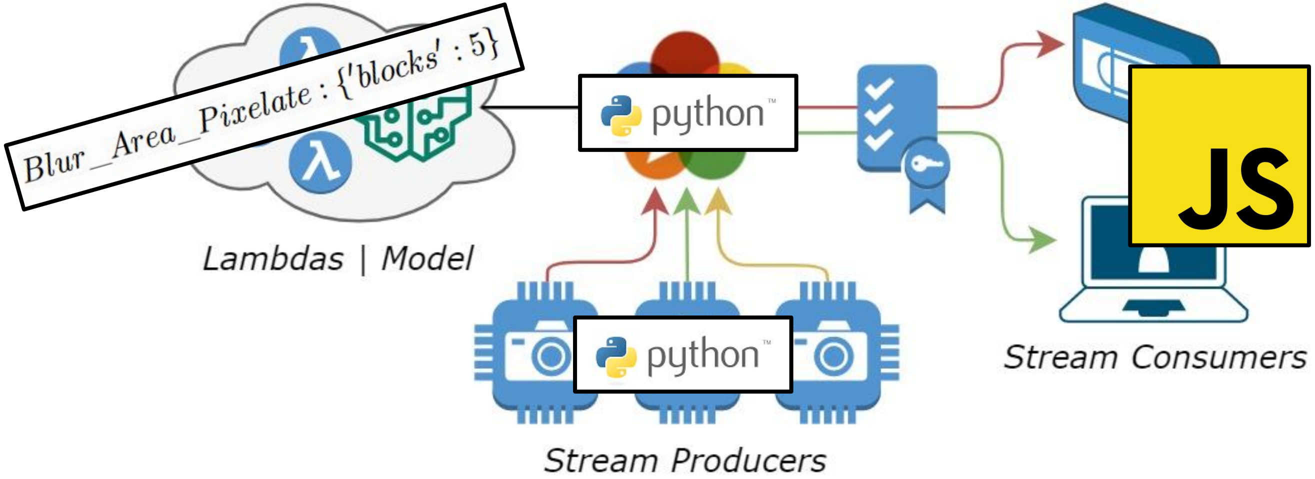


IoT device
(stream producer)

Abstract Concept - Architecture

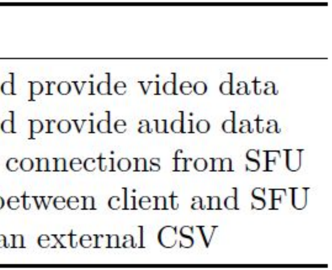


Prototype - Video Streaming

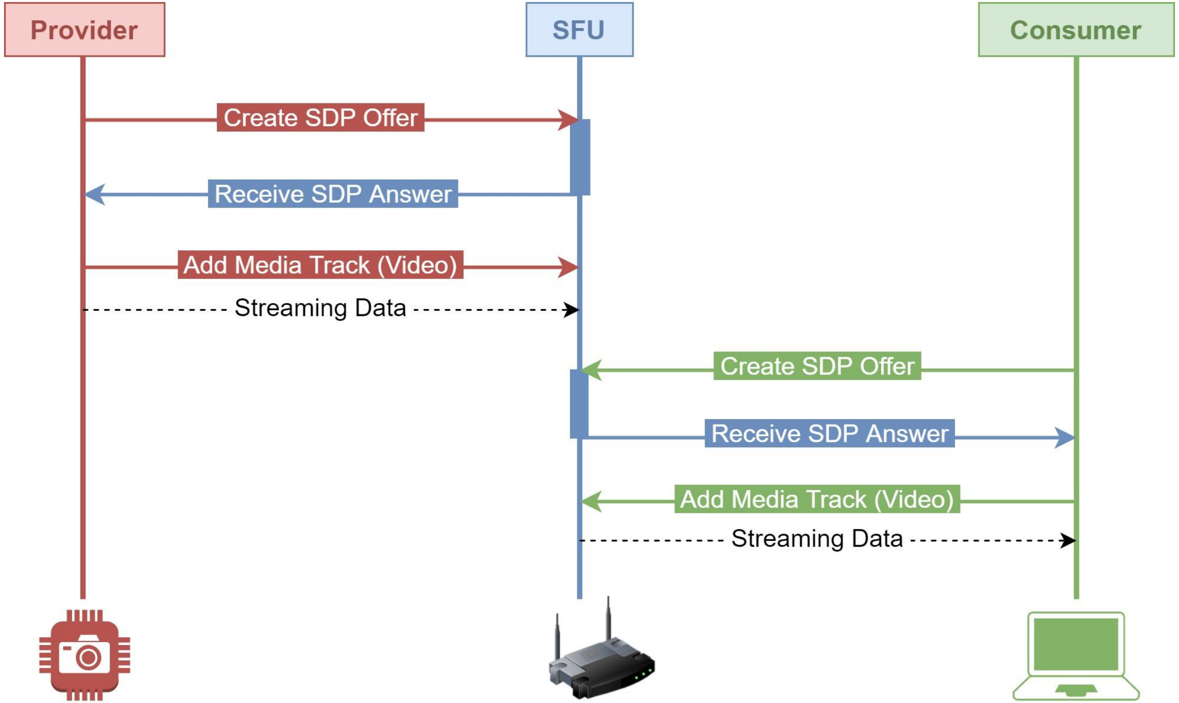


https://en.logodownload.org/python-logo/ https://commons.wikimedia.org/wiki/File:Unofficial_JavaScript_logo_2.svg

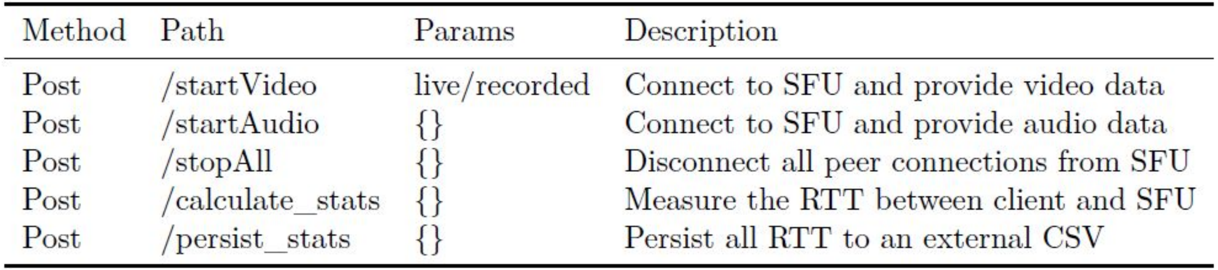
Prototype - Data Provision & Consumption (1/2)



d provide video data
d provide audio data
connections from SFU
between client and SFU
an external CSV

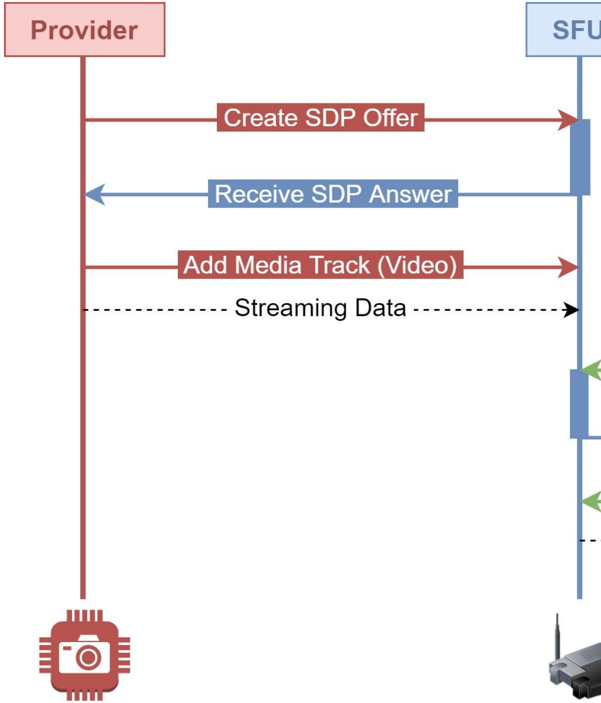


Prototype - Data Provision & Consumption (2/2)

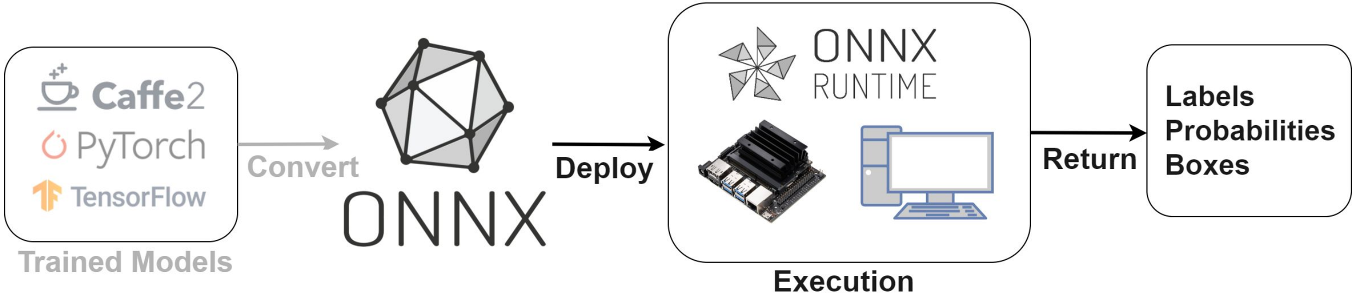


Method	Path	Params	Description
Post	/startVideo	live/recorded	Connect to SFU and provide video data
Post	/startAudio	{}	Connect to SFU and provide audio data
Post	/stopAll	{}	Disconnect all peer connections from SFU
Post	/calculate_stats	{}	Measure the RTT between client and SFU
Post	/persist_stats	{}	Persist all RTT to an external CSV

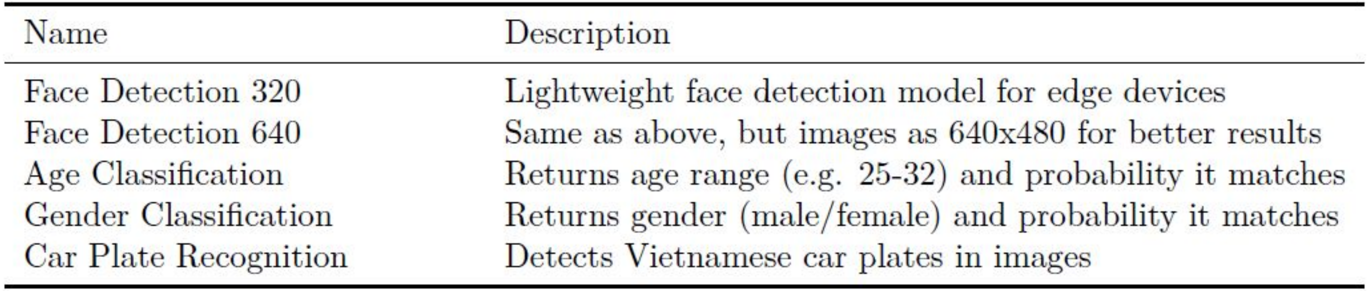
REST interface for Python client



Prototype - Pattern Detection (1/2)



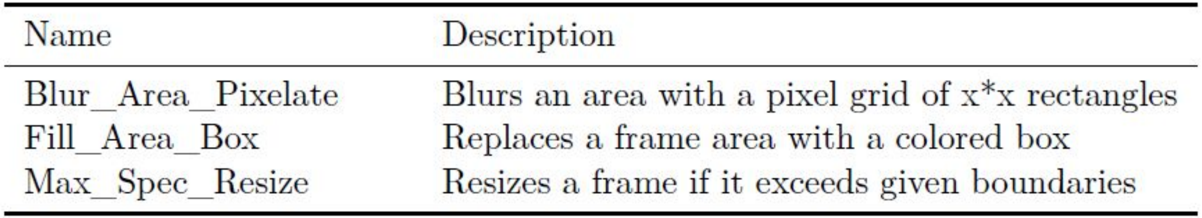
Prototype - Pattern Detection (2/2)



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

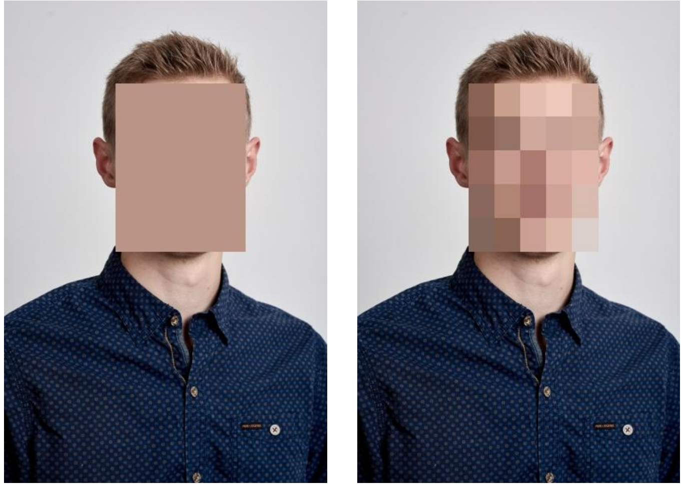
Prototype - Transformation Functions



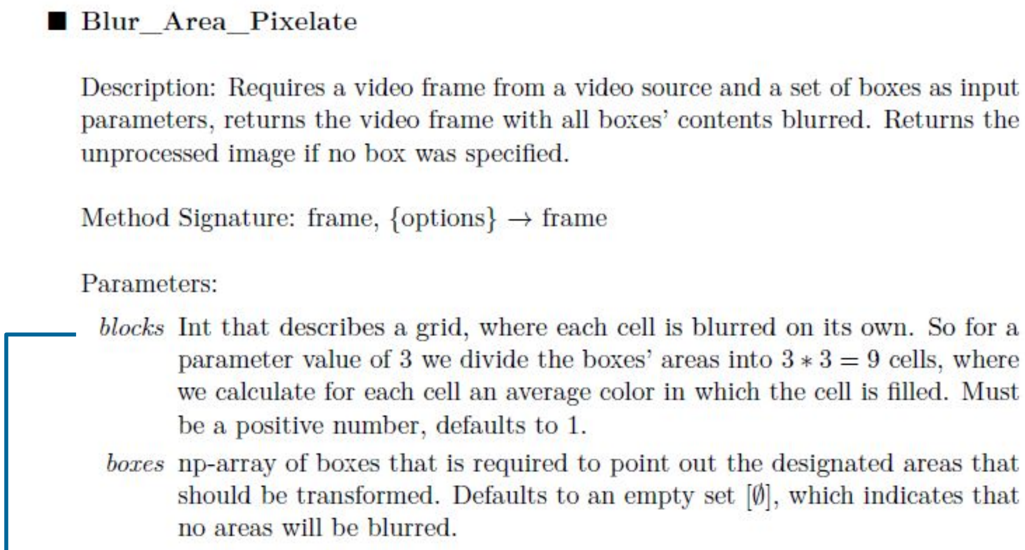
Name	Description
Blur_Area_Pixelate	Blurs an area with a pixel grid of x*x rectangles
Fill_Area_Box	Replaces a frame area with a colored box
Max_Spec_Resize	Resizes a frame if it exceeds given boundaries

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.

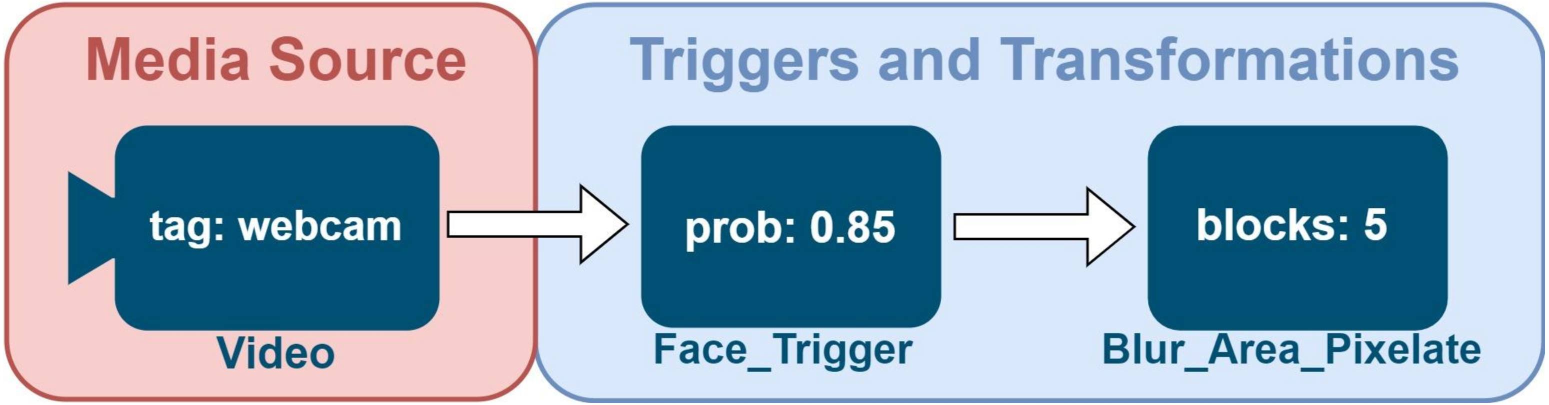
  Method Signature: frame, {options} → frame

  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.

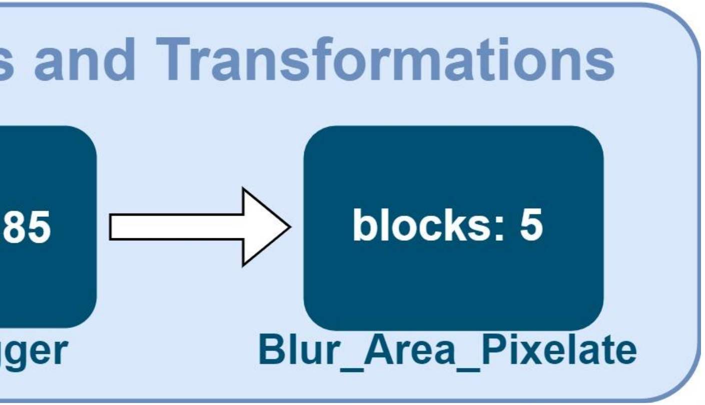
  boxes np-array of boxes that is required to point out the designated areas that should be transformed. Defaults to an empty set [$\emptyset$], which indicates that no areas will be blurred.

Prototype - Privacy Model / Chain



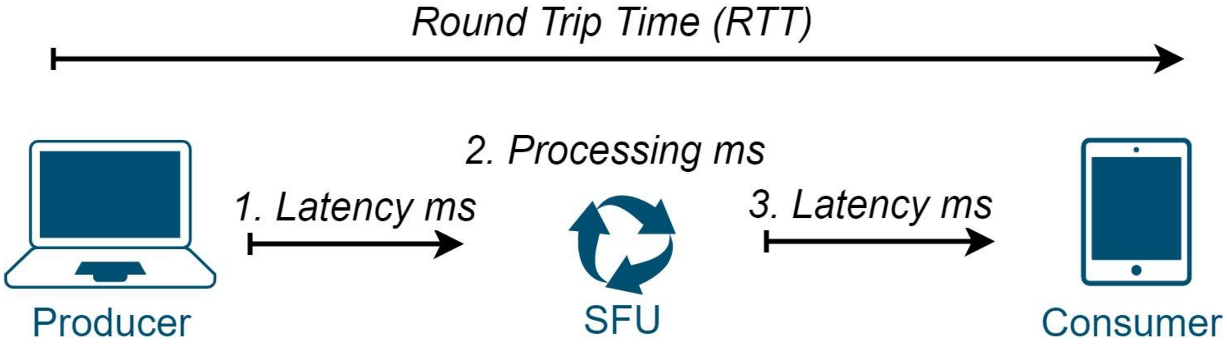
$$video : \{'tag' :' webcam'\} \rightarrow Face\_Trigger : \{'prob' : 0.85\} \rightarrow Blur\_Area\_Pixelate : \{'blocks' : 5\}$$

Prototype - Demo

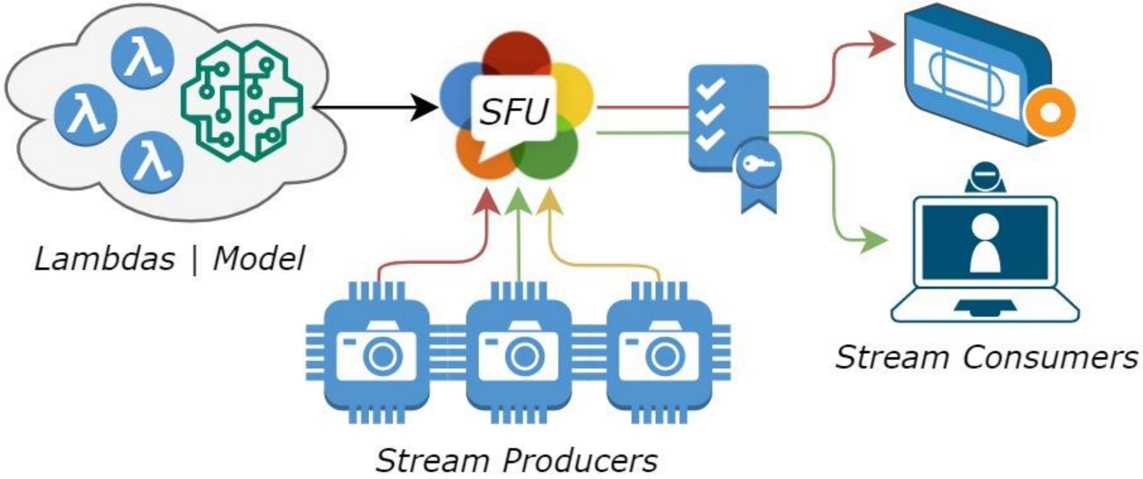


$$5\} \rightarrow Blur\_Area\_Pixelate : \{'blocks' : 5\}$$

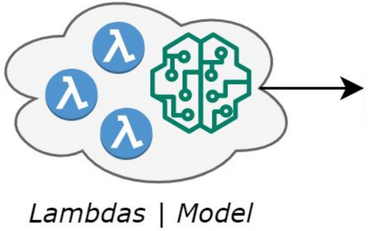
Evaluation - Metrics

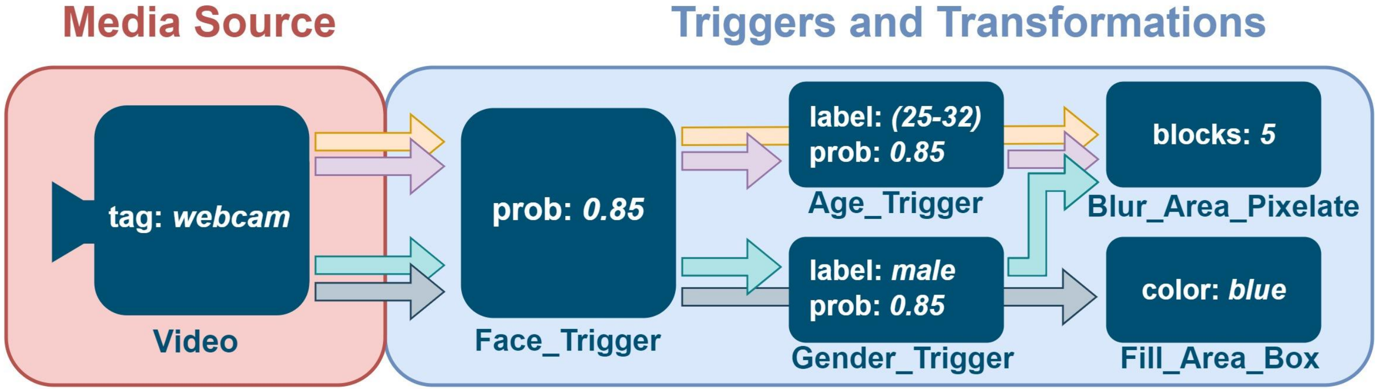


Evaluation - Variation (1/5)



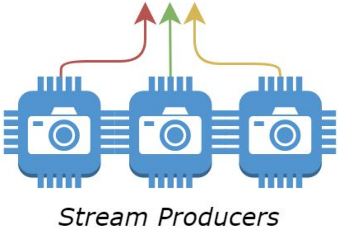
Evaluation - Variation (2/5)

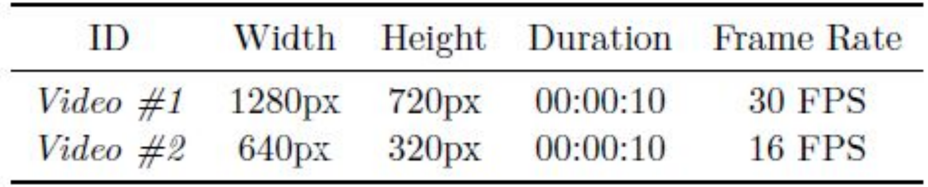




Privacy models used for evaluation

Evaluation - Variation (3/5)





ID	Width	Height	Duration	Frame Rate
Video #1	1280px	720px	00:00:10	30 FPS
Video #2	640px	320px	00:00:10	16 FPS

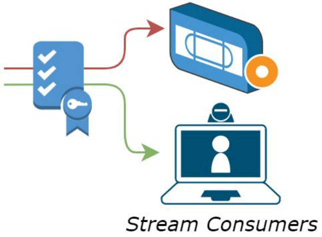
Streamed video quality & FPS

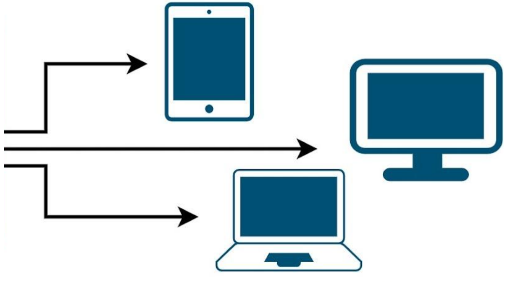
Evaluation - Variation (4/5)

Edge vs Cloud environment

https://de.m.wikipedia.org/wiki/Datei:Amazon_Web_Services_Logo.svg

Evaluation - Variation (5/5)

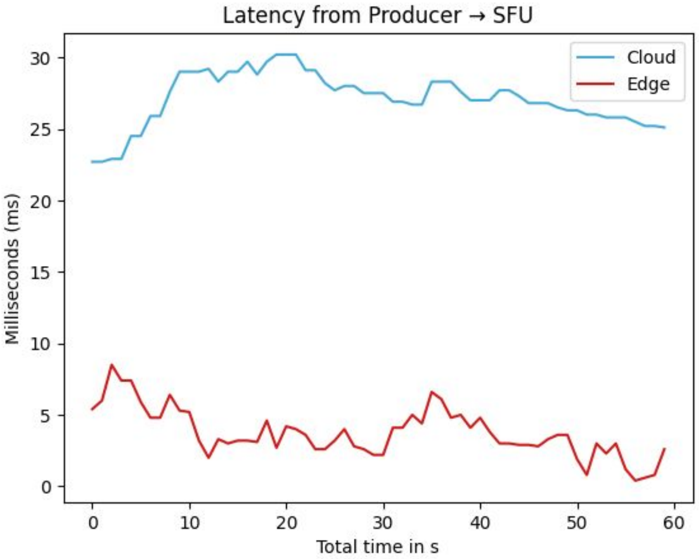




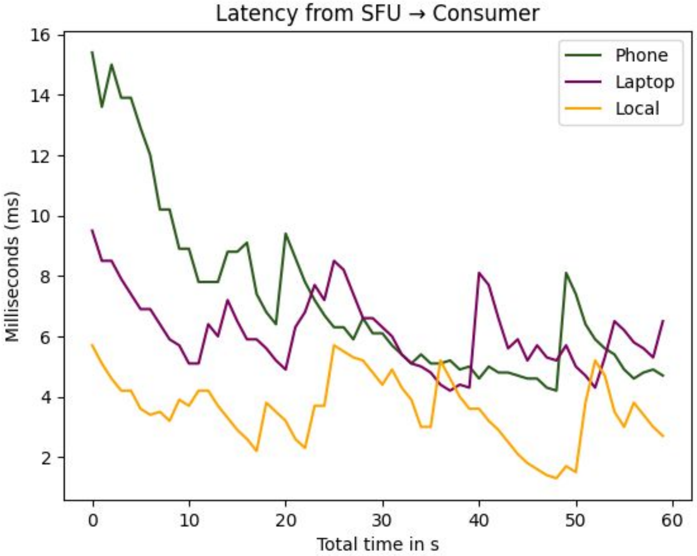
Laptop, Smartphone, or Local

https://de.m.wikipedia.org/wiki/Datei:Amazon_Web_Services_Logo.svg

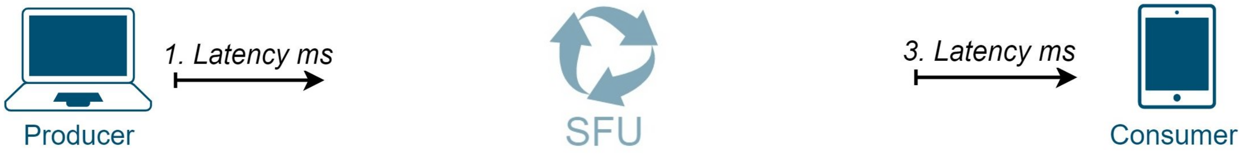
Results - Producer / Consumer



Latency from Producer → SFU



Latency from SFU → Consumer



1. Latency ms

Producer

SFU

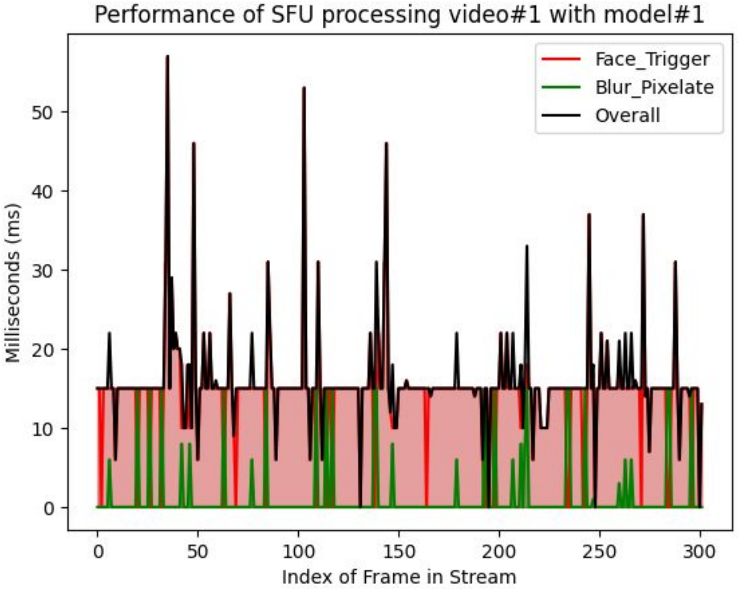
3. Latency ms

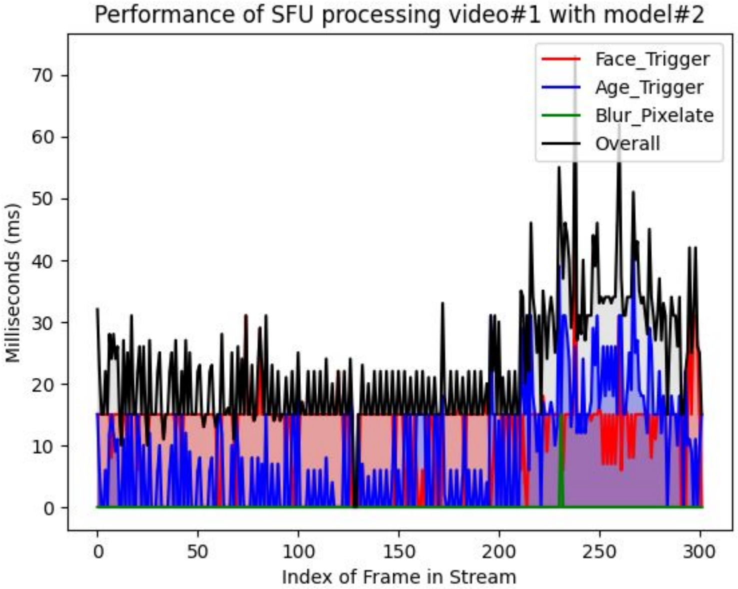
Consumer

Results - SFU (1/3)

2. Processing ms

SFU

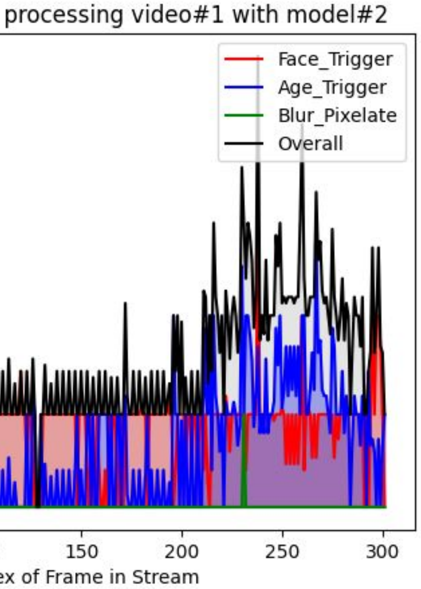




Results - SFU (2/3)

2. Processing ms

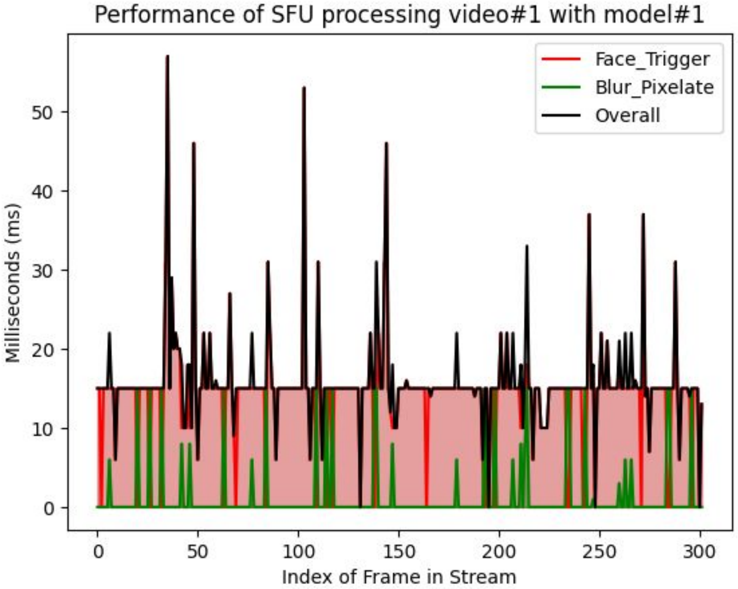
SFU

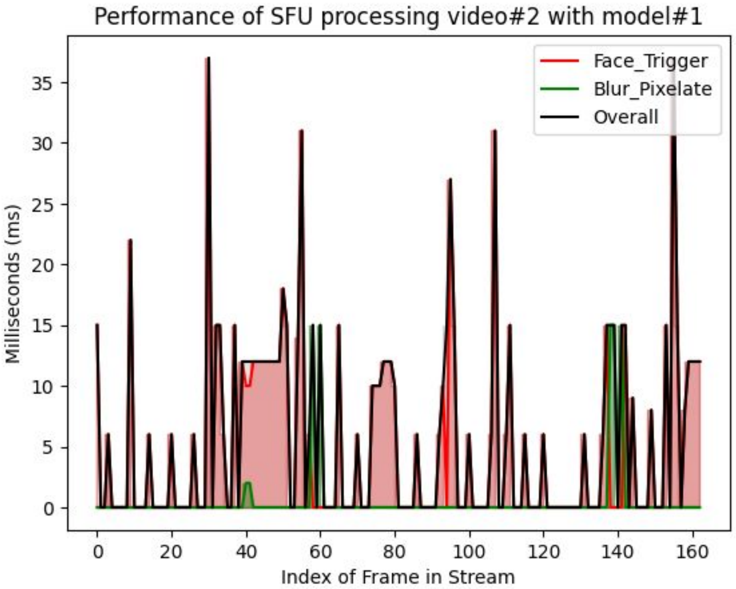


Results - SFU (3/3)

2. Processing ms

SFU





Conclusion

- Proof of concept provided
    - Privacy model specification
    - Model enforcement on video streams
- Latency to SFU decreased by moving to edge
    - Producer - SFU: 28ms → 4ms
- Resulting latency fulfills requirements
    - 15ms face detection → 60 FPS streaming

Conclusion - Future Work

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

Conclusion - Outlook

IEEE ICFEC 2022

—

IEEE Access

Thank you for your attention!
Questions?