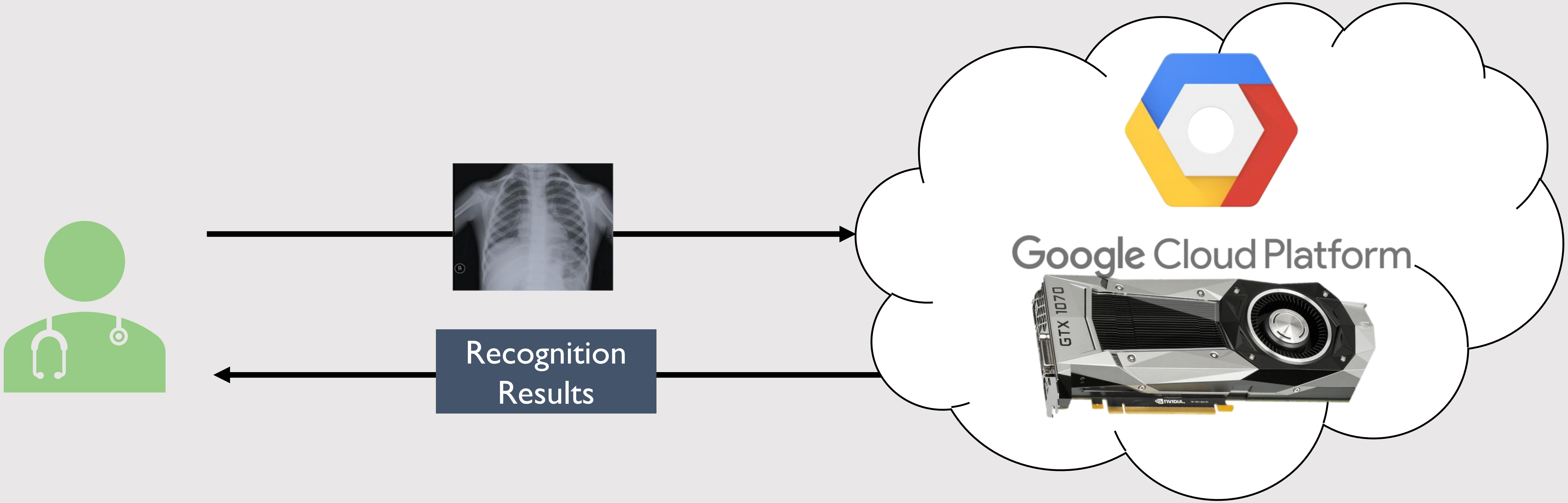
Telekine: Secure Computing with Cloud GPUs

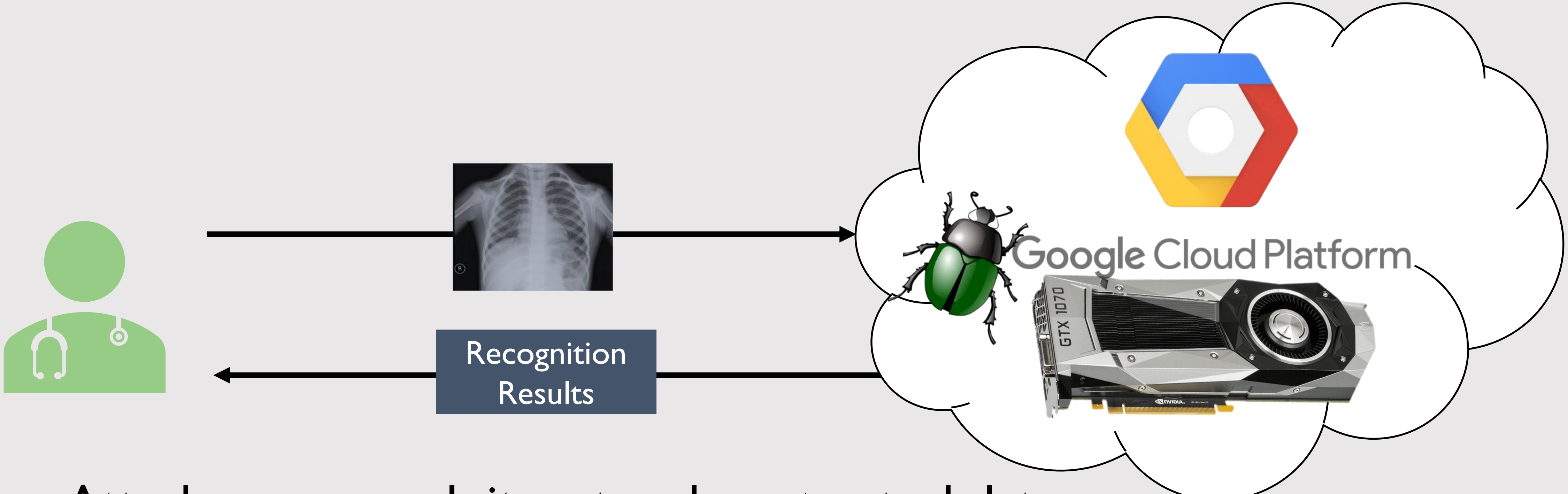
NSDI 2020

Tyler Hunt, Zhipeng Jia, Vance Miller, Ariel Szekely, Yige Hu,
Christopher J. Rossbach, Emmett Witchel

Trusting the cloud provider is difficult

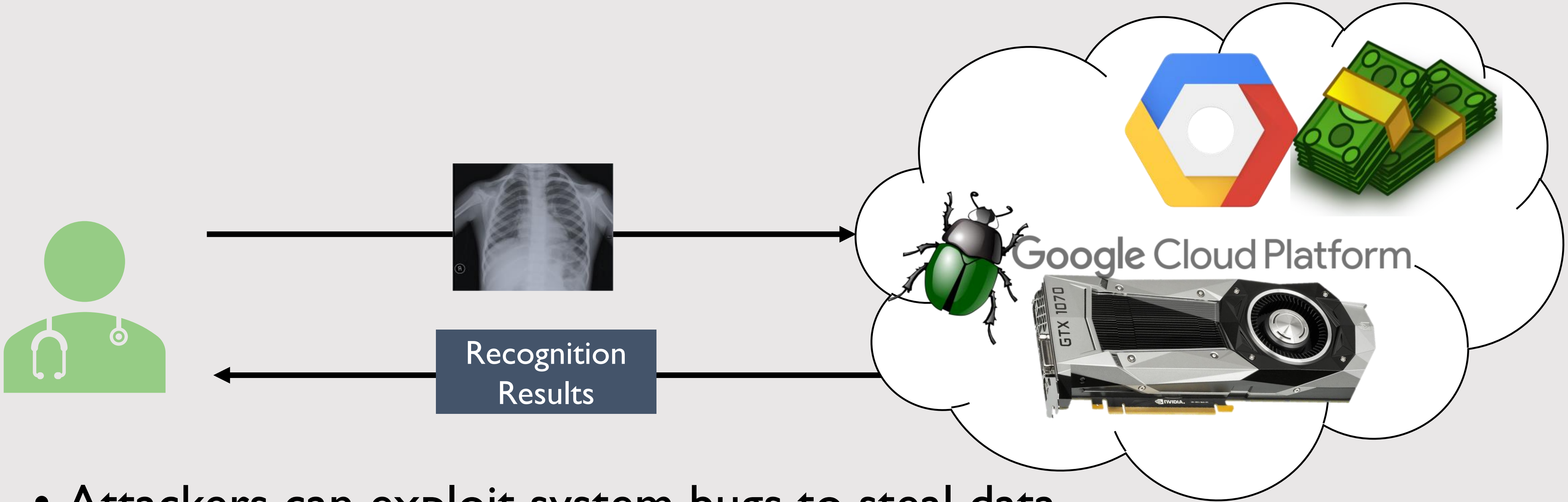


Trusting the cloud provider is difficult



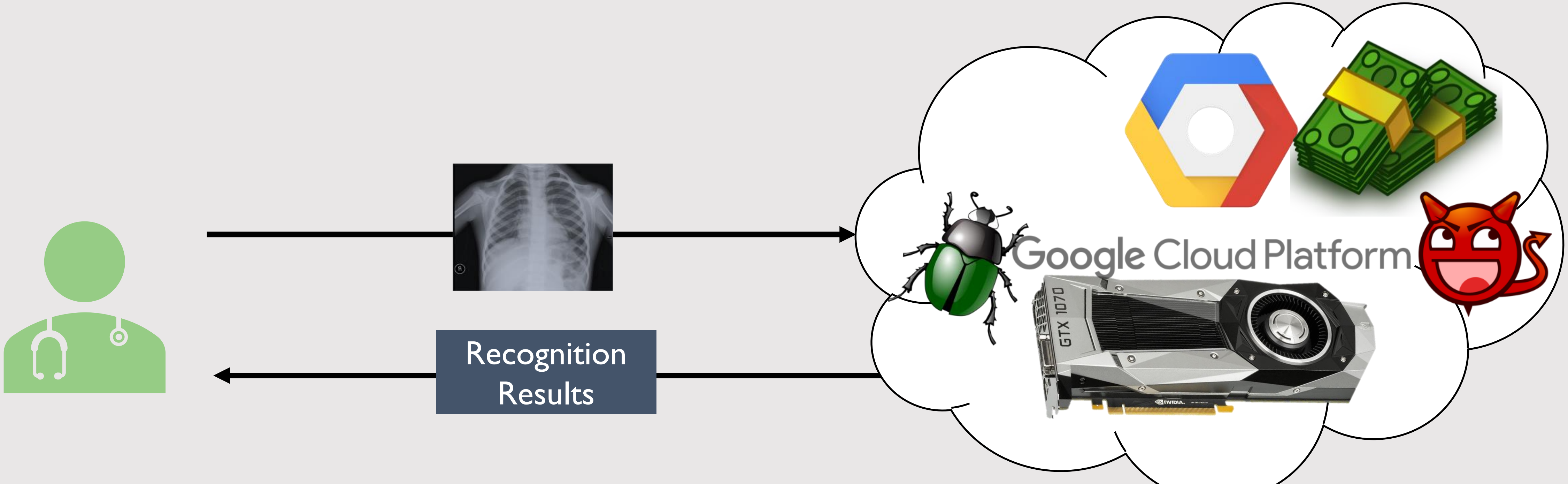
- Attackers can exploit system bugs to steal data

Trusting the cloud provider is difficult



- Attackers can exploit system bugs to steal data
- The cloud provider has their own interests (e.g., monetizing user data)

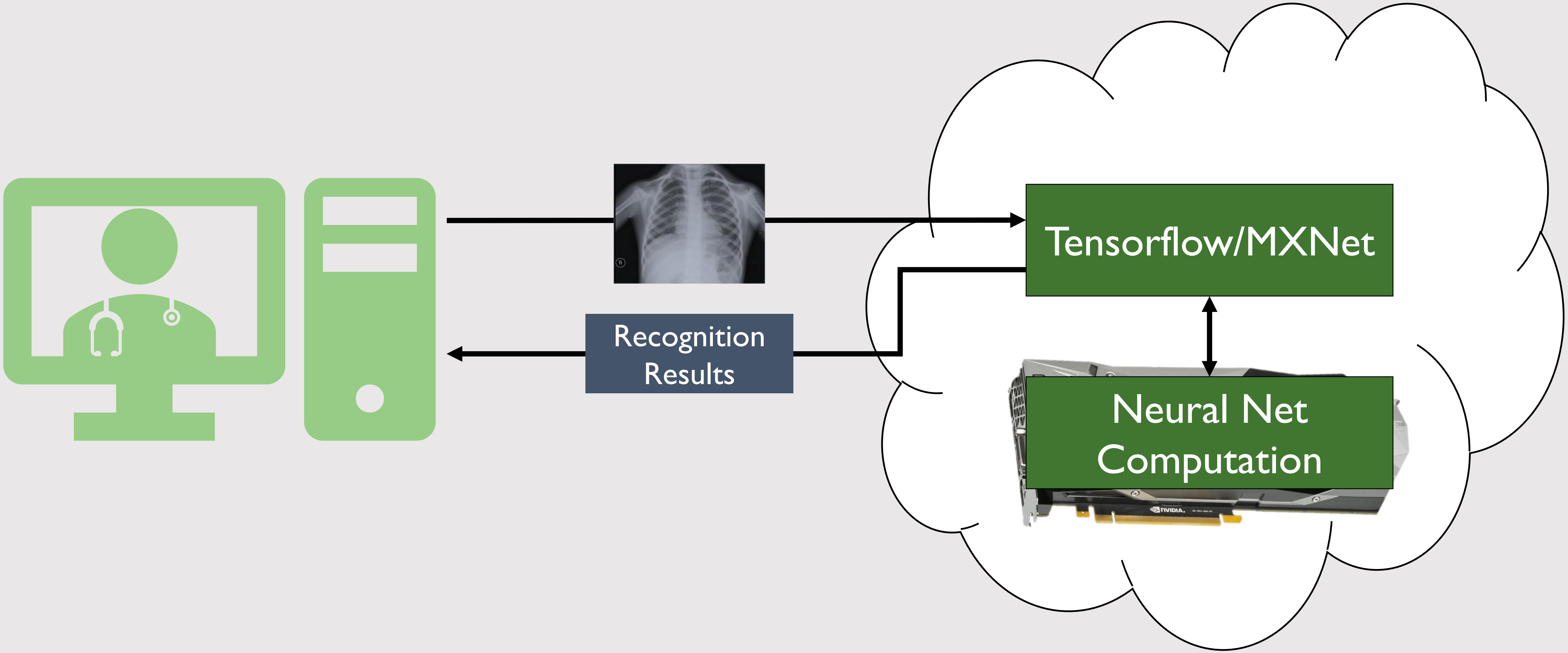
Trusting the cloud provider is difficult



- Attackers can exploit system bugs to steal data
- The cloud provider has their own interests (e.g., monetizing user data)
- Many administrators; some may be malicious

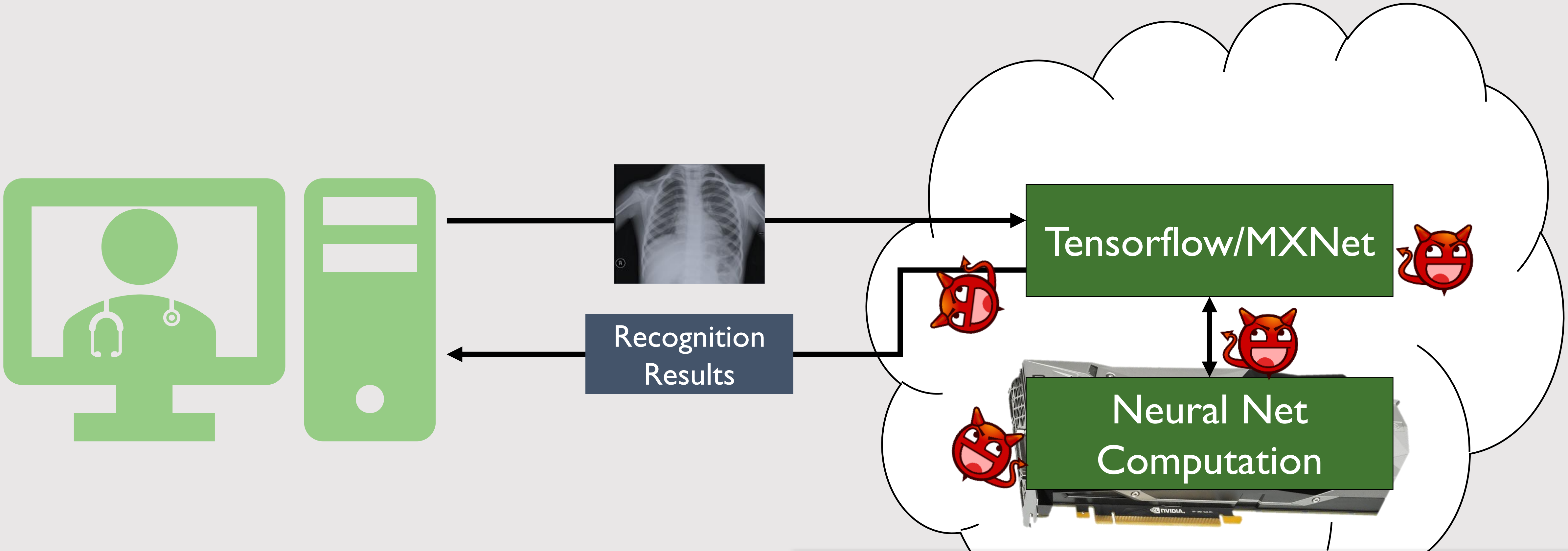
Legend: Trusted | Untrusted | Data | Trusted

Avoiding trust in the cloud provider is difficult



Legend: Trusted | Untrusted | Data
Trusted

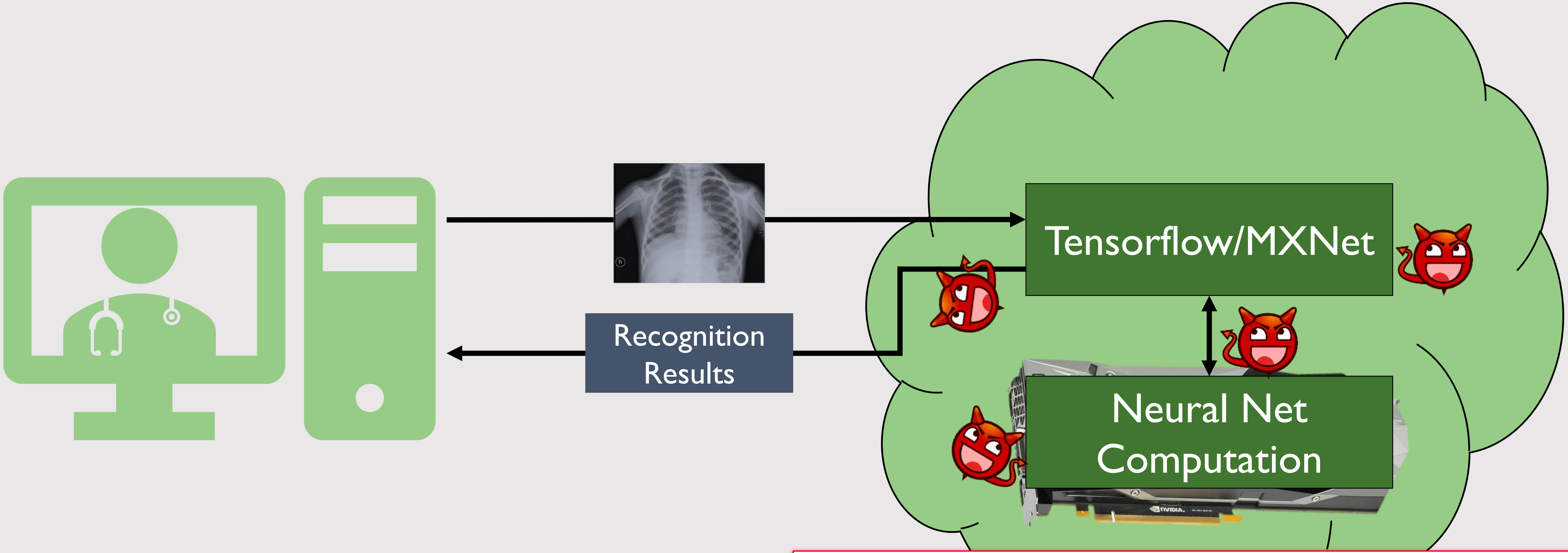
Avoiding trust in the cloud provider is difficult



OS/Hypervisor allows provider to see user's secret data

Legend: Trusted | Untrusted | Data
Trusted

Avoiding trust in the cloud provider is difficult



OS/Hypervisor allows provider to see user's secret data

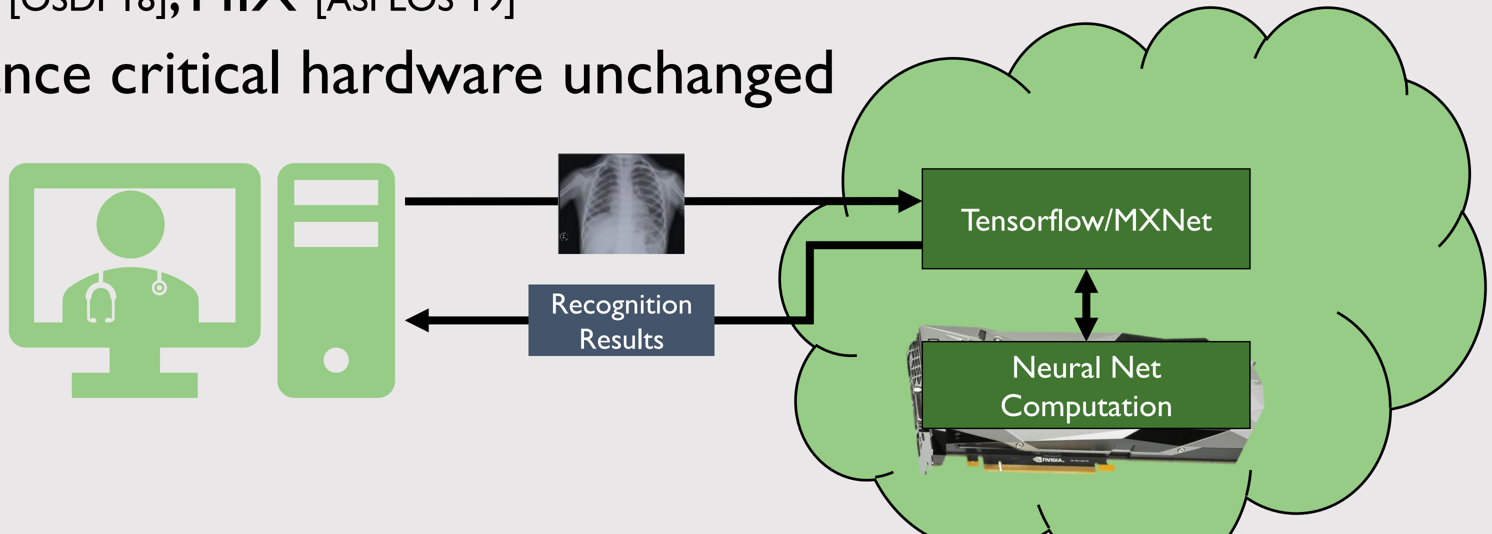
Legend:

Trusted	Untrusted	Data
Trusted		

Introduce TEEs to isolate computation

(TEE is Trusted Execution Environment)

- TEEs cannot be bypassed by software
  - Hardware root of trust (e.g., SGX on Intel, TrustZone on ARM)
- Protect communication from the provider with cryptography
- Research proposals exist for GPU TEEs
  - Graviton [OSDI`18], HIX [ASPLOS`19]
  - Performance critical hardware unchanged

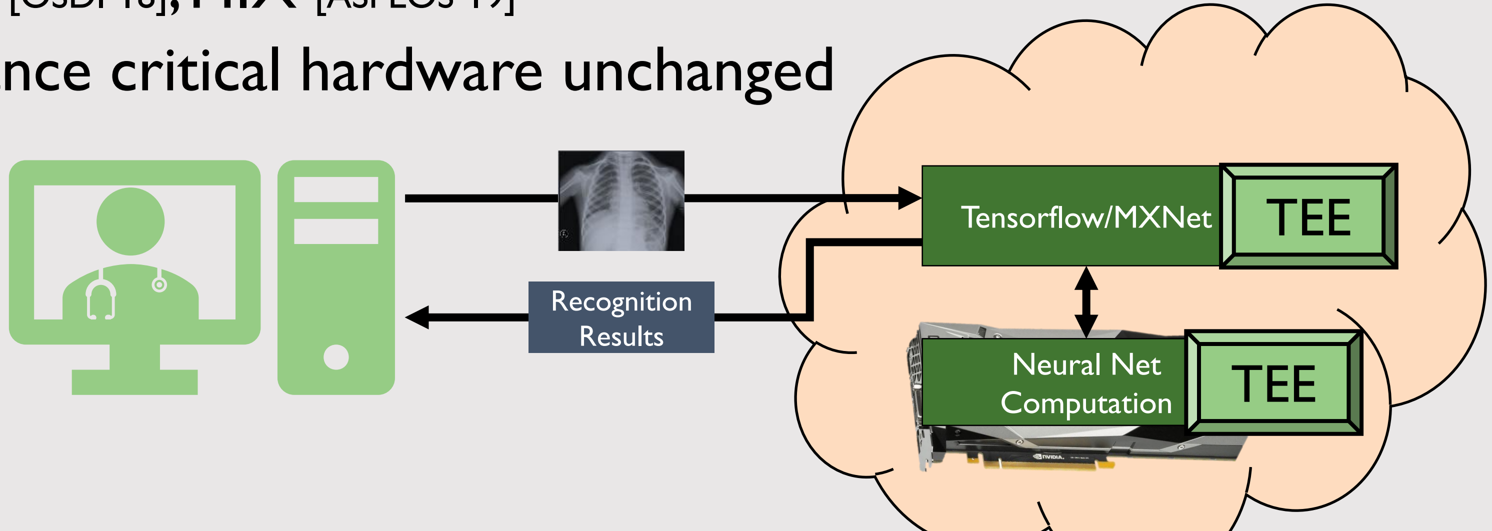


Legend: Trusted | Untrusted | Data
Trusted

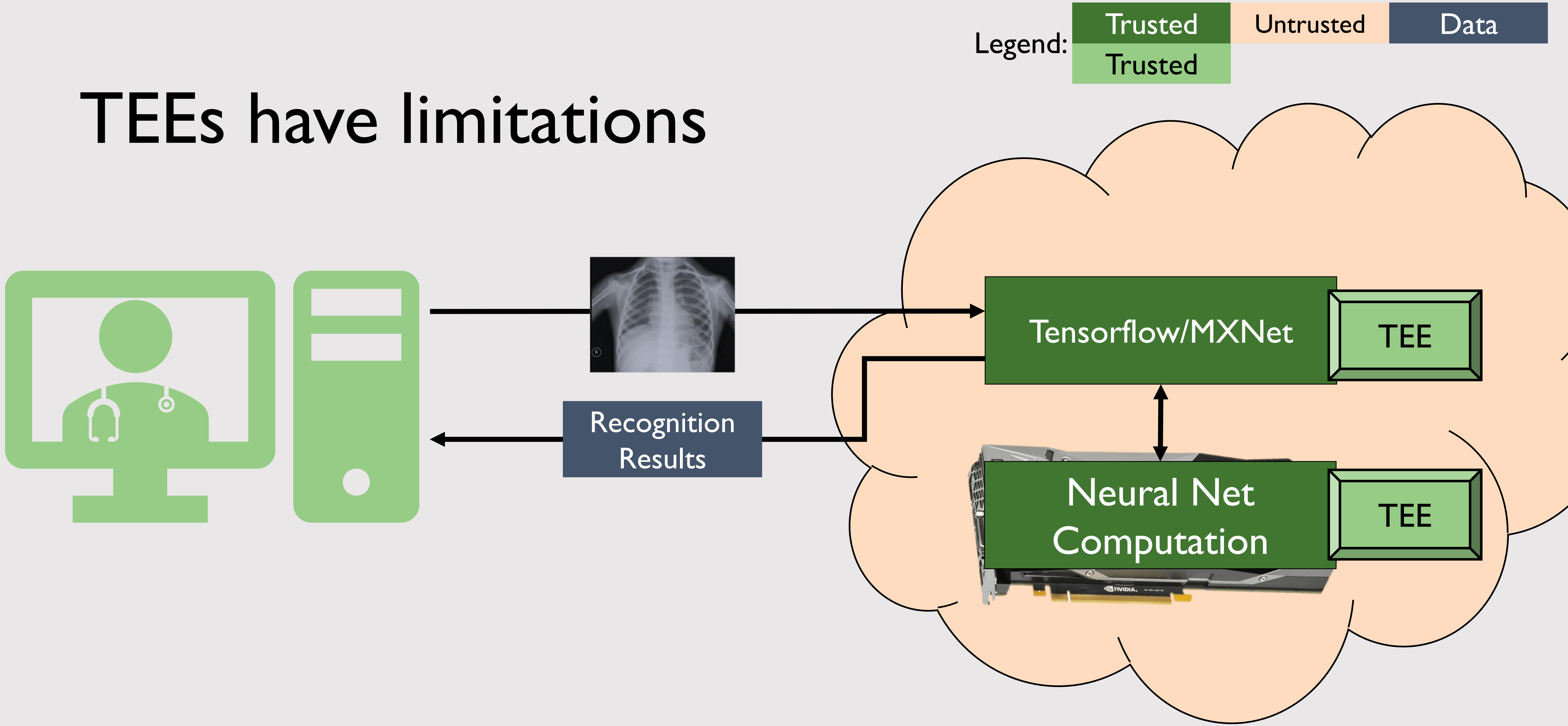
Introduce TEEs to isolate computation

(TEE is Trusted Execution Environment)

- TEEs cannot be bypassed by software
  - Hardware root of trust (e.g., SGX on Intel, TrustZone on ARM)
- Protect communication from the provider with cryptography
- Research proposals exist for GPU TEEs
  - Graviton [OSDI`18], HIX [ASPLOS`19]
  - Performance critical hardware unchanged

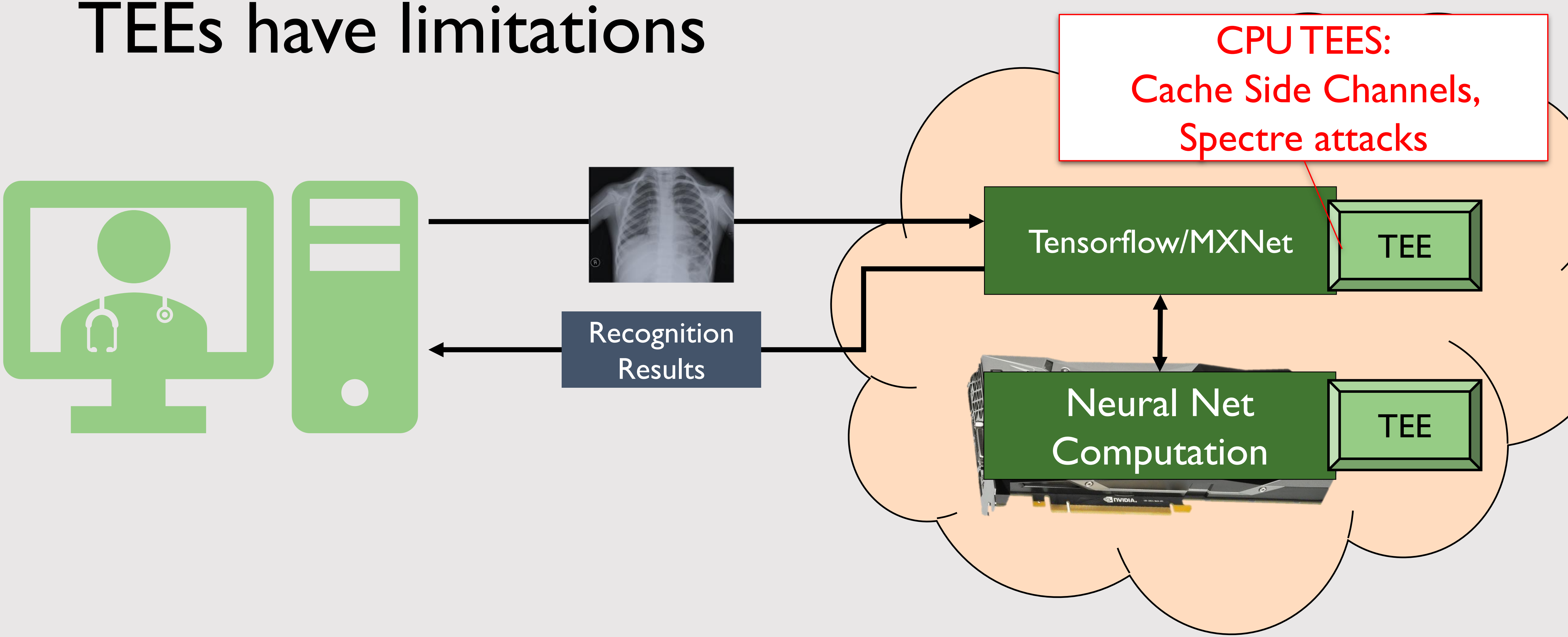


TEEs have limitations



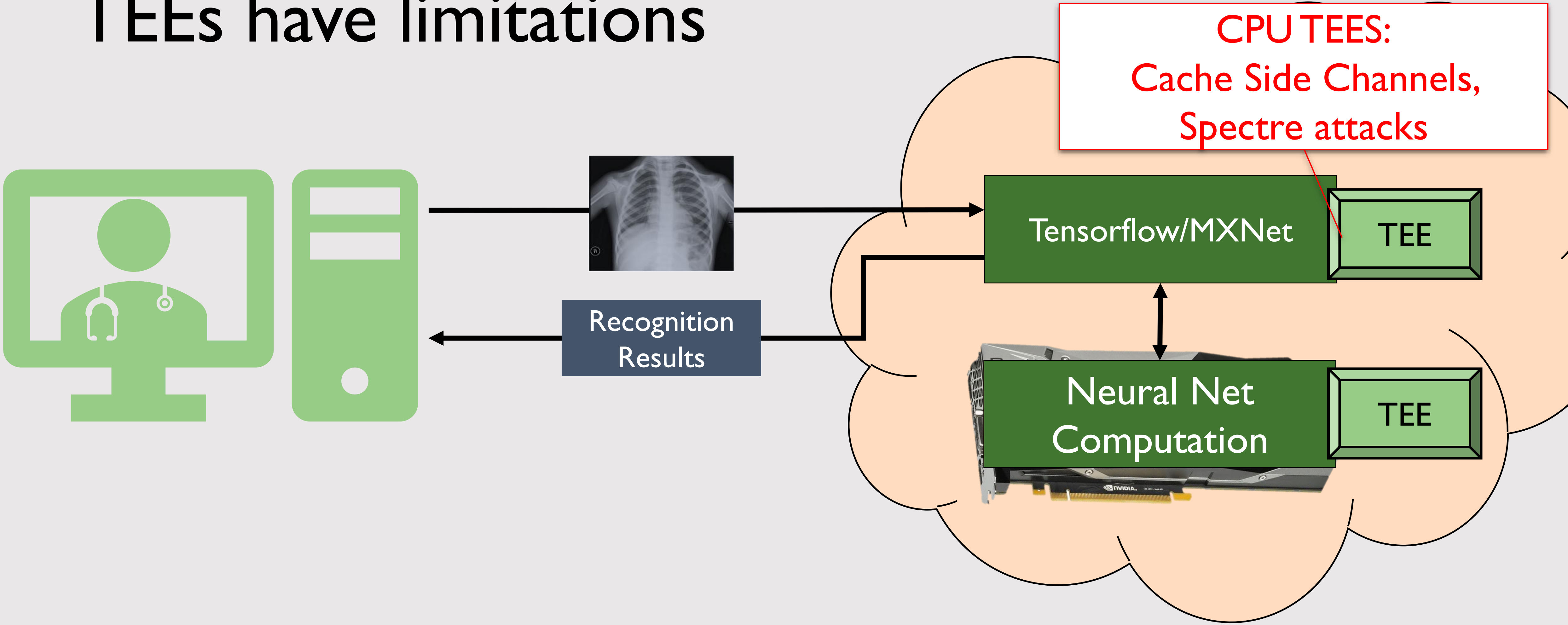
Legend: Trusted | Untrusted | Data | Trusted

TEEs have limitations



TEEs have limitations

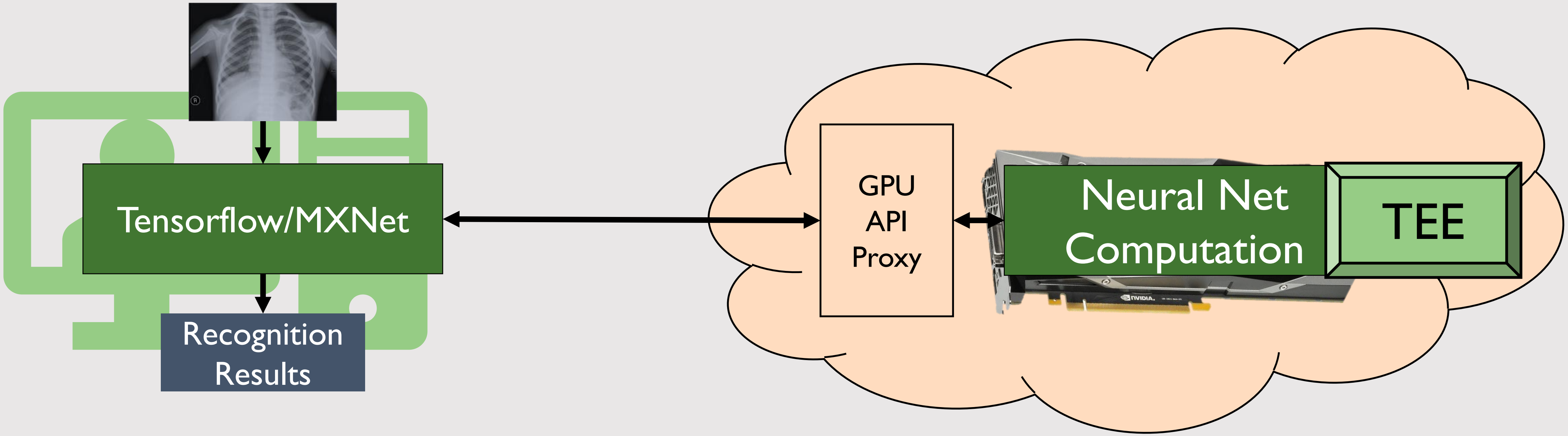
Legend: Trusted | Untrusted | Data
Trusted



Mitigations require heroic effort, especially for complex software

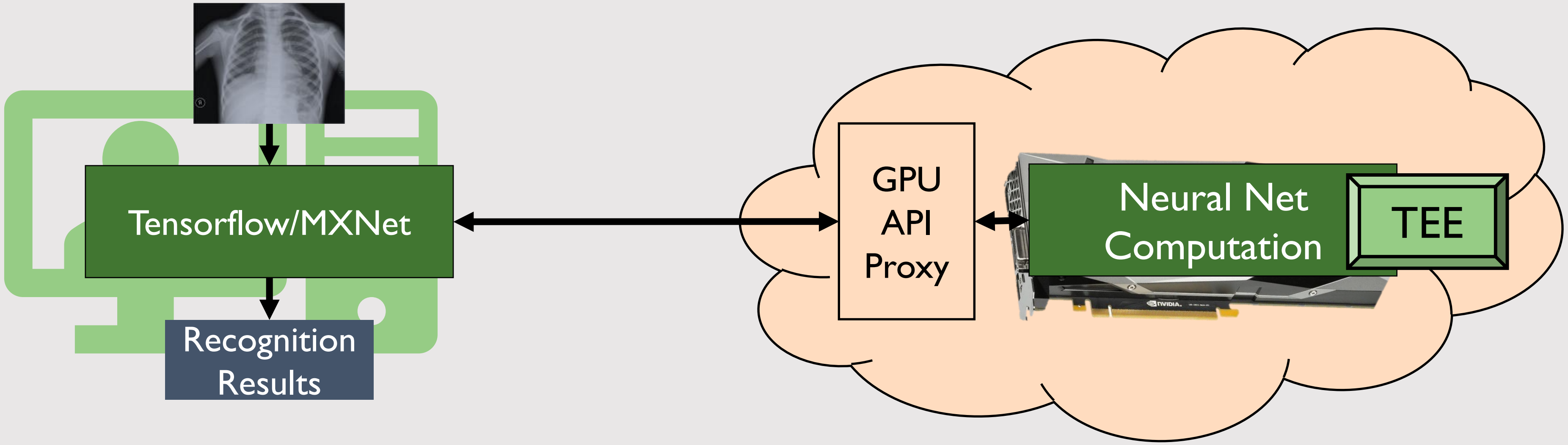
Legend: Trusted | Trusted | Untrusted | Data

Telekine addresses TEE limitations

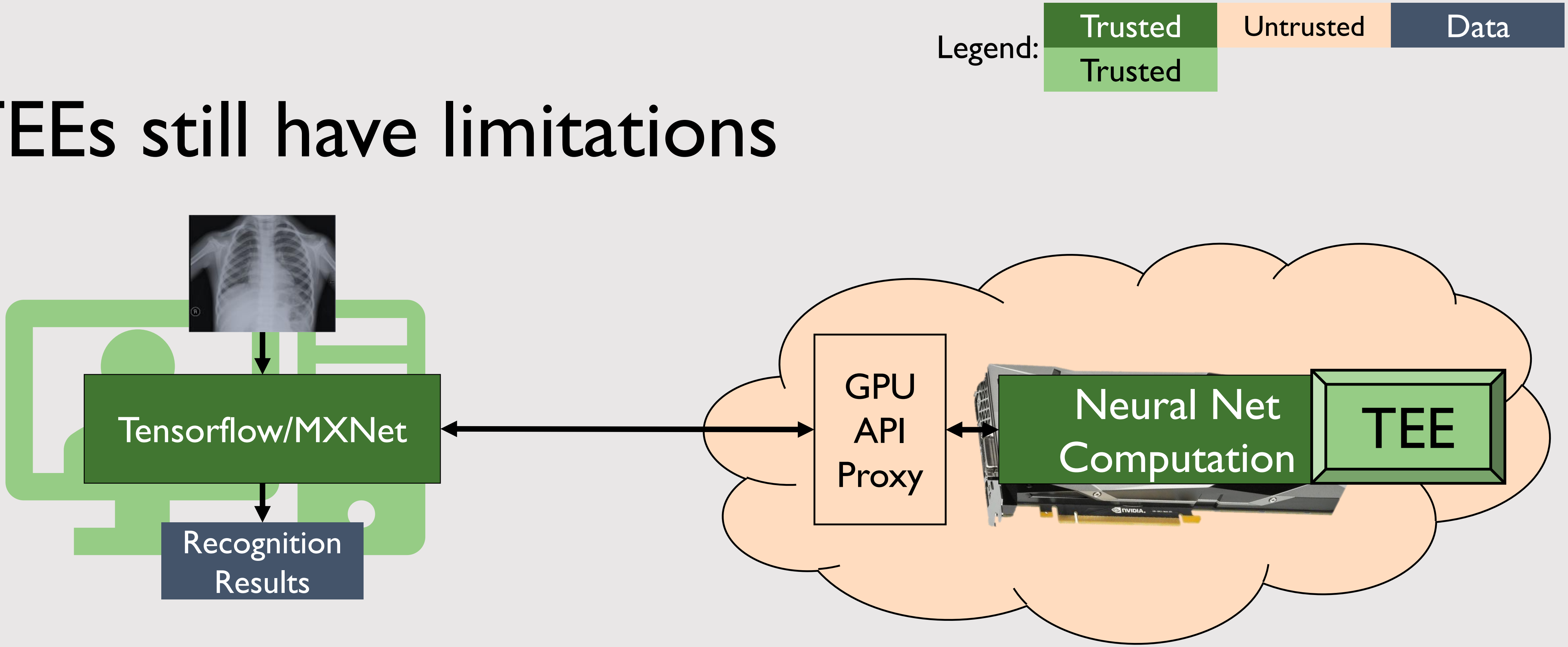


Telekine uses API-remoting instead of CPU TEEs

- Interpose on GPU API calls
- Application does not have to be modified, user does not need GPU
- Turn every API call into an RPC, executed by the remote machine
- Traffic is encrypted/authenticated; the proxy does not need protection

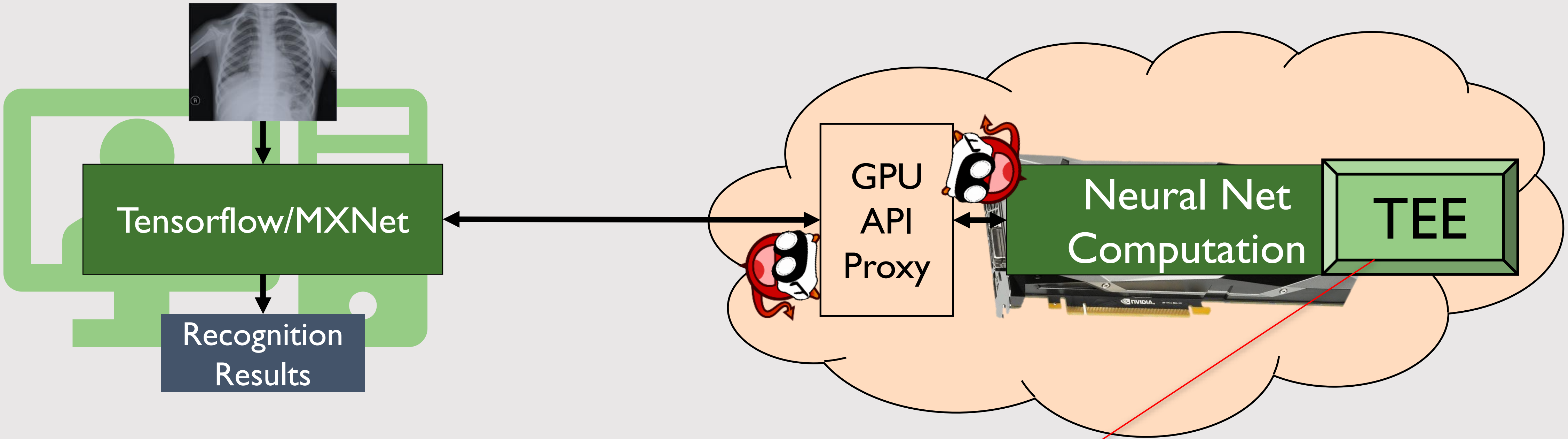


TEEs still have limitations



Legend: Trusted | Trusted | Untrusted | Data

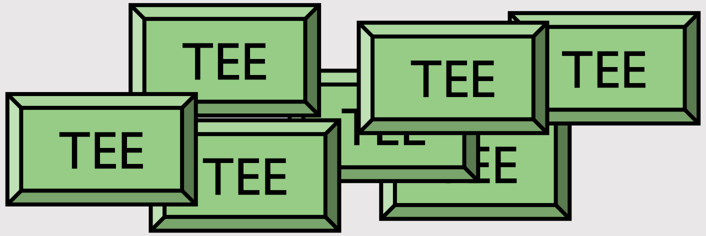
TEEs still have limitations



All TEES:
Communication timing attacks

Telekine addresses communication timing channels

- TEEs do not consider communication side channels
  - Securing the processor (CPU/GPU) does not secure communication
- GPU programing paradigm features frequent communication
  - CPU-to-CPU communication is also vulnerable
- Communication patterns tend to leak timing information
  - E.g., GPU kernel execution time



In the rest of the talk we will answer:

- Can information be extracted from GPU communication patterns?
- How does Telekine remove that information?
- What are Telekine’s overheads?

In the rest of the talk we will answer:

- Can information be extracted from GPU communication patterns?
  Yes, we demonstrate a communication timing attack
- How does Telekine remove that information?
- What are Telekine's overheads?

In the rest of the talk we will answer:

- Can information be extracted from GPU communication patterns?

  Yes, we demonstrate a communication timing attack

- How does Telekine remove that information?

  Replace GPU streams with new data-oblivious streams

- What are Telekine's overheads?

In the rest of the talk we will answer:

- Can information be extracted from GPU communication patterns?

  Yes, we demonstrate a communication timing attack

- How does Telekine remove that information?

  Replace GPU streams with new data-oblivious streams

- What are Telekine's overheads?

  Overheads are reasonable: ~20% for neural network training

In the rest of the talk we will answer:

- Can information be extracted from GPU communication patterns?

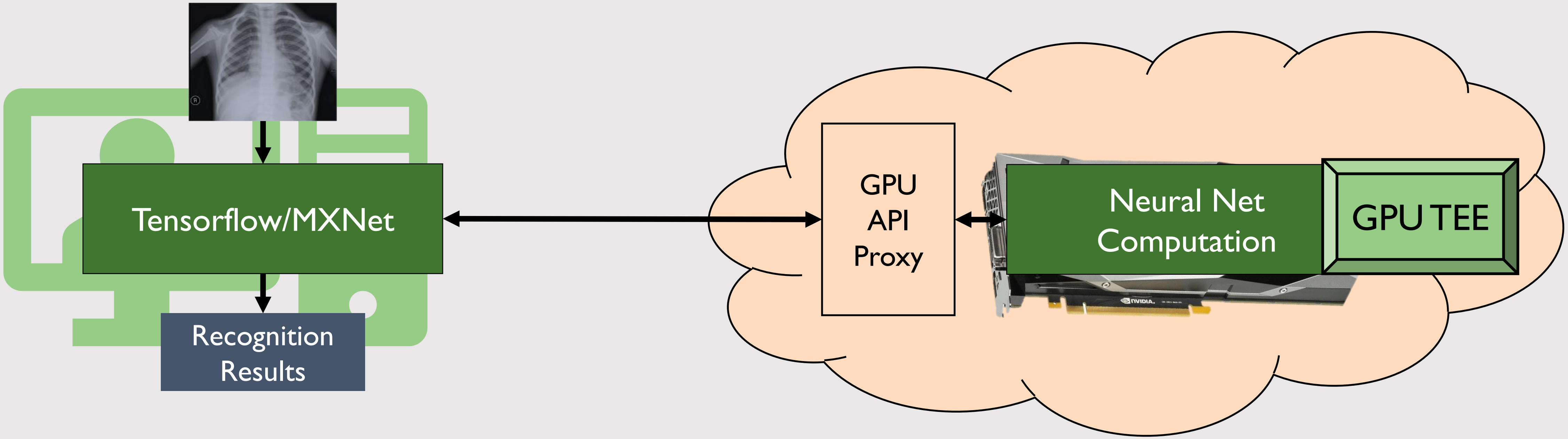
  Yes, we demonstrate a communication timing attack
- How does Telekine remove that information?

  Replace GPU streams with new data-oblivious streams
- What are Telekine's overheads?

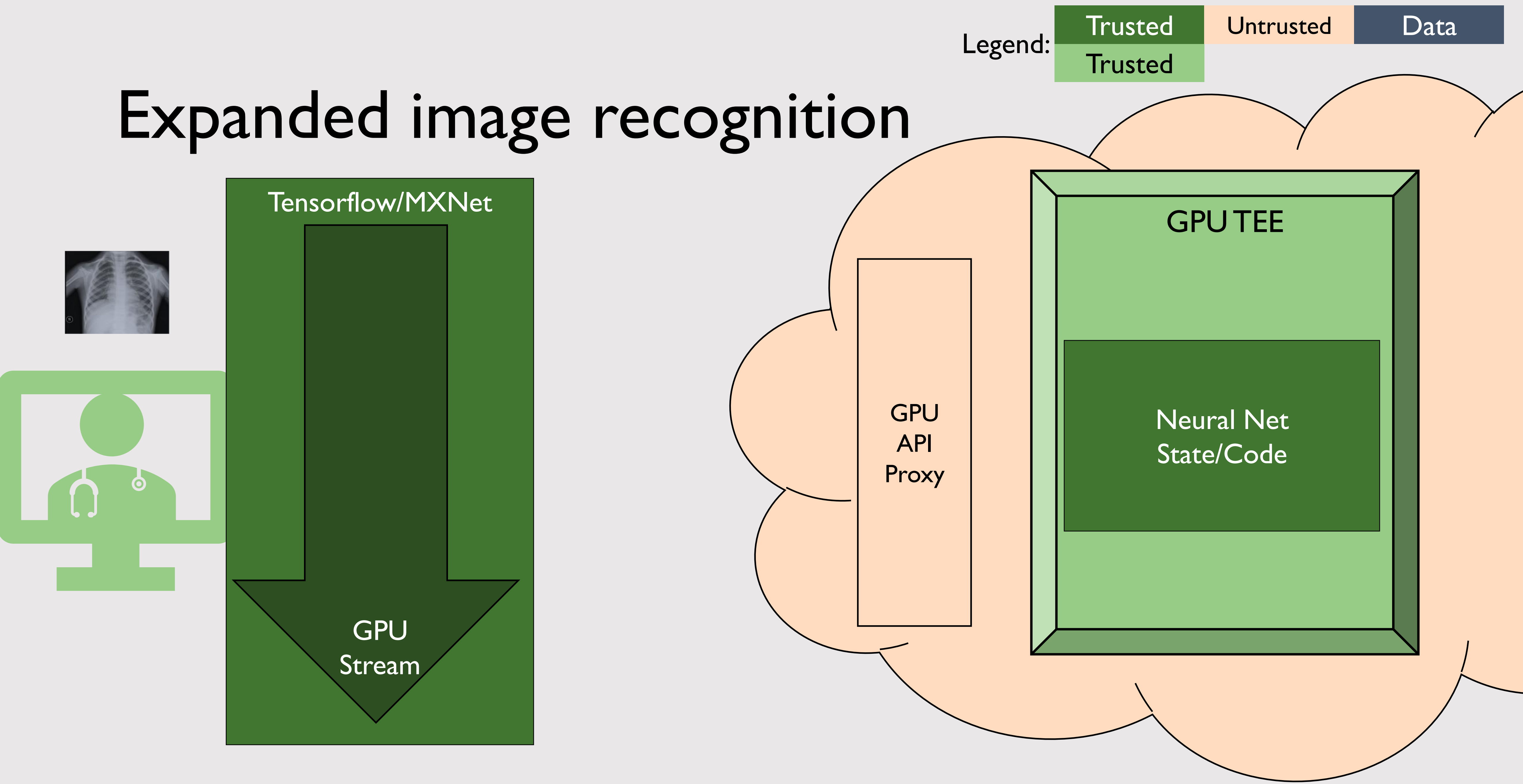
  Overheads are reasonable: ~20% for neural network training

Legend: Trusted | Trusted | Untrusted | Data

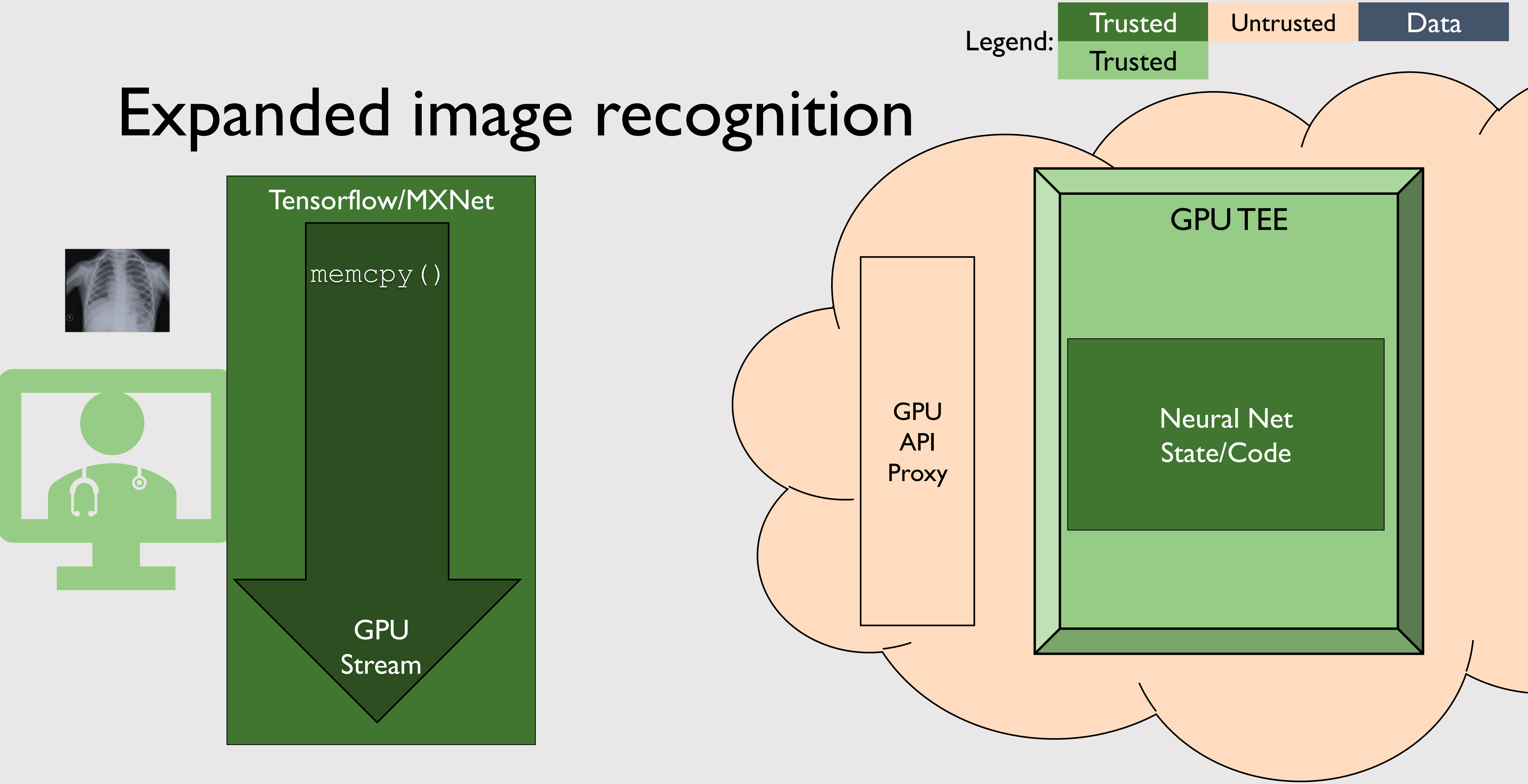
Expanded image recognition



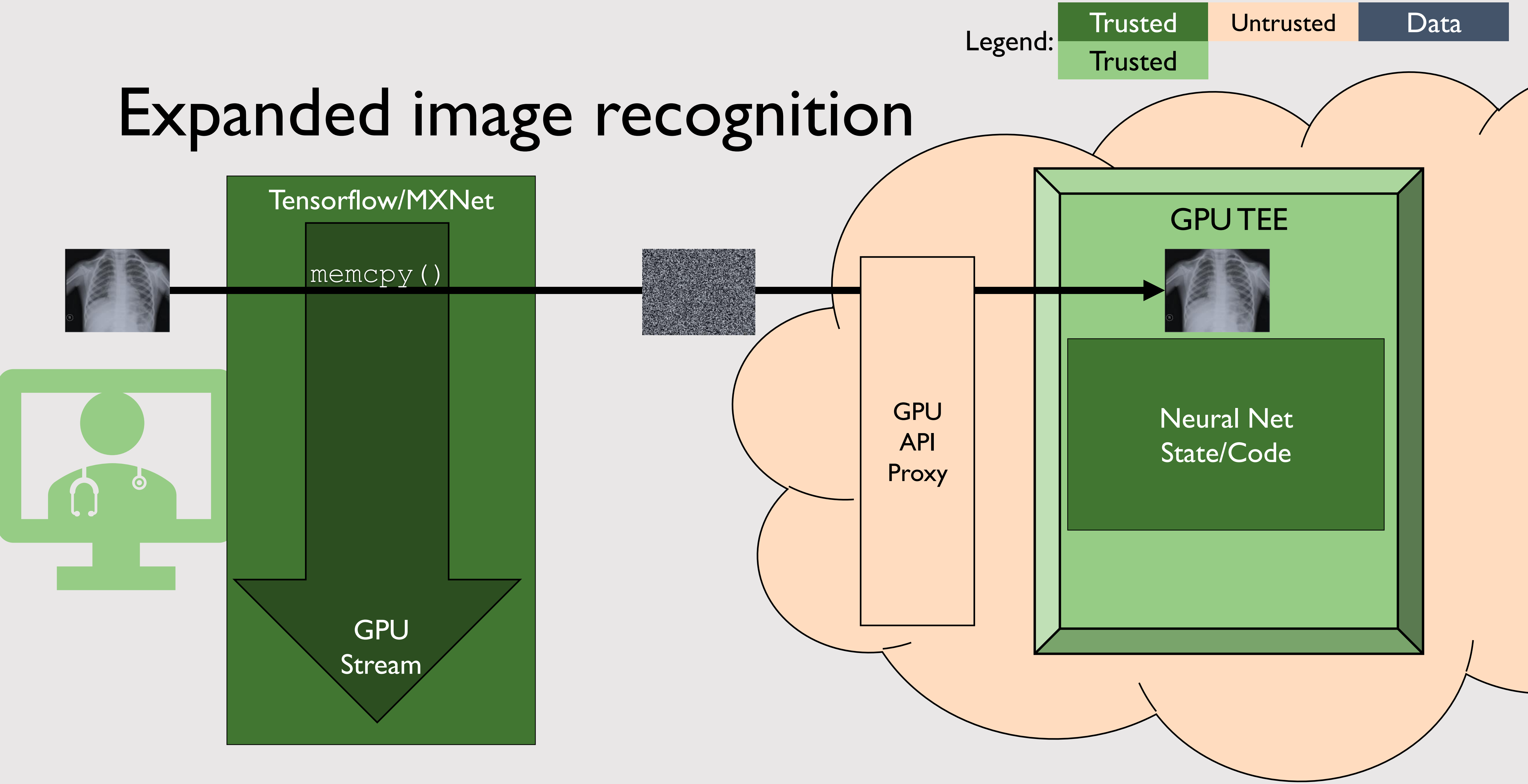
Expanded image recognition



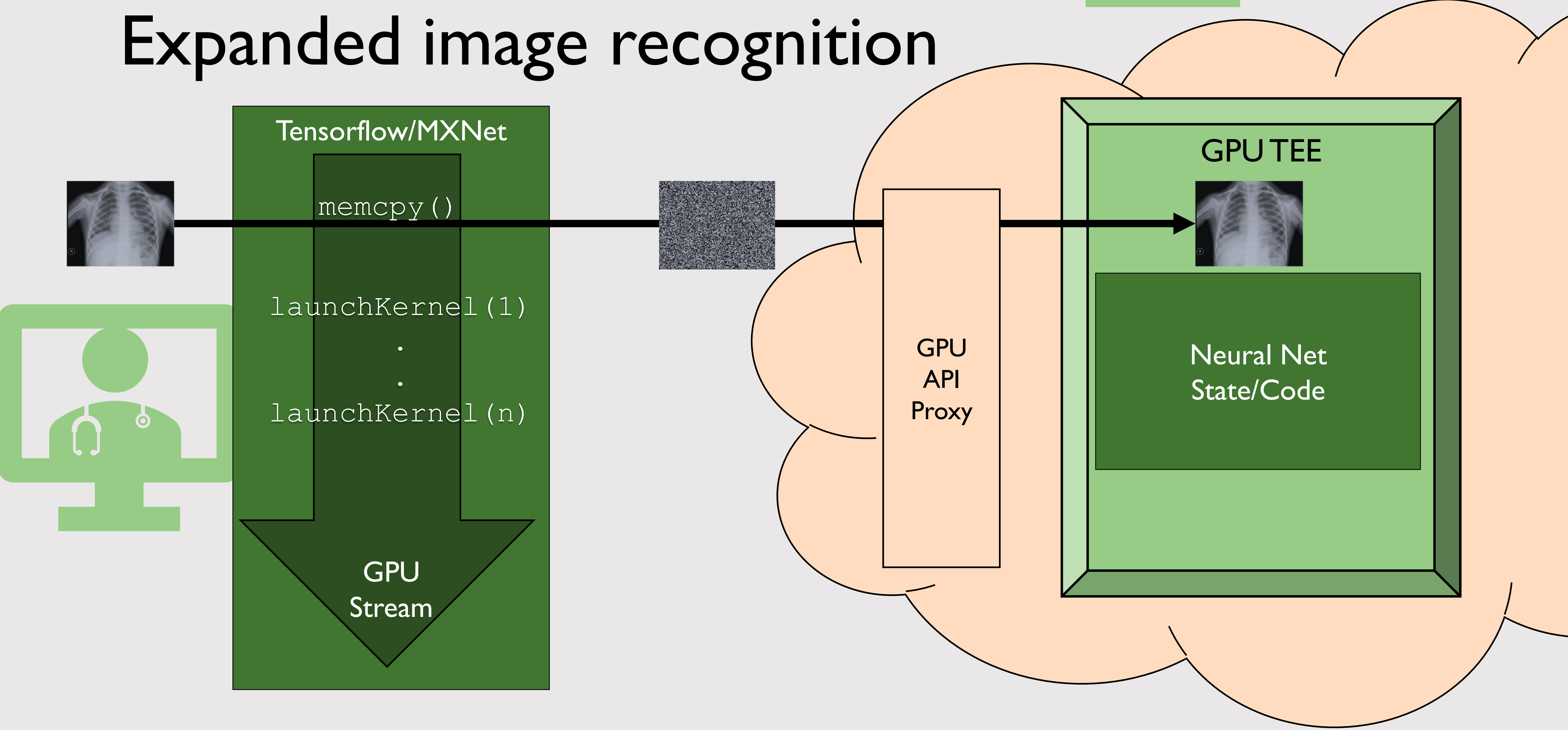
Expanded image recognition



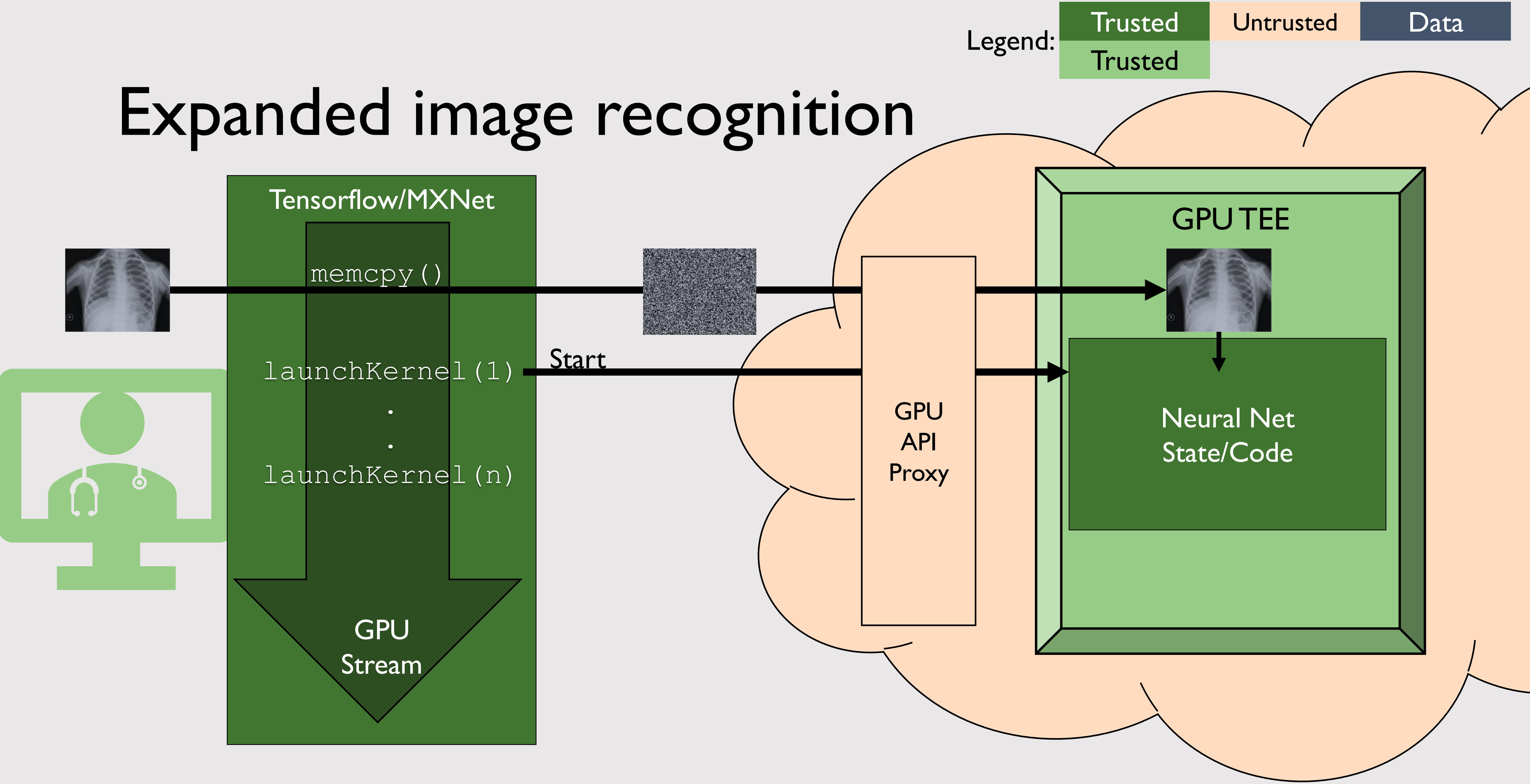
Expanded image recognition



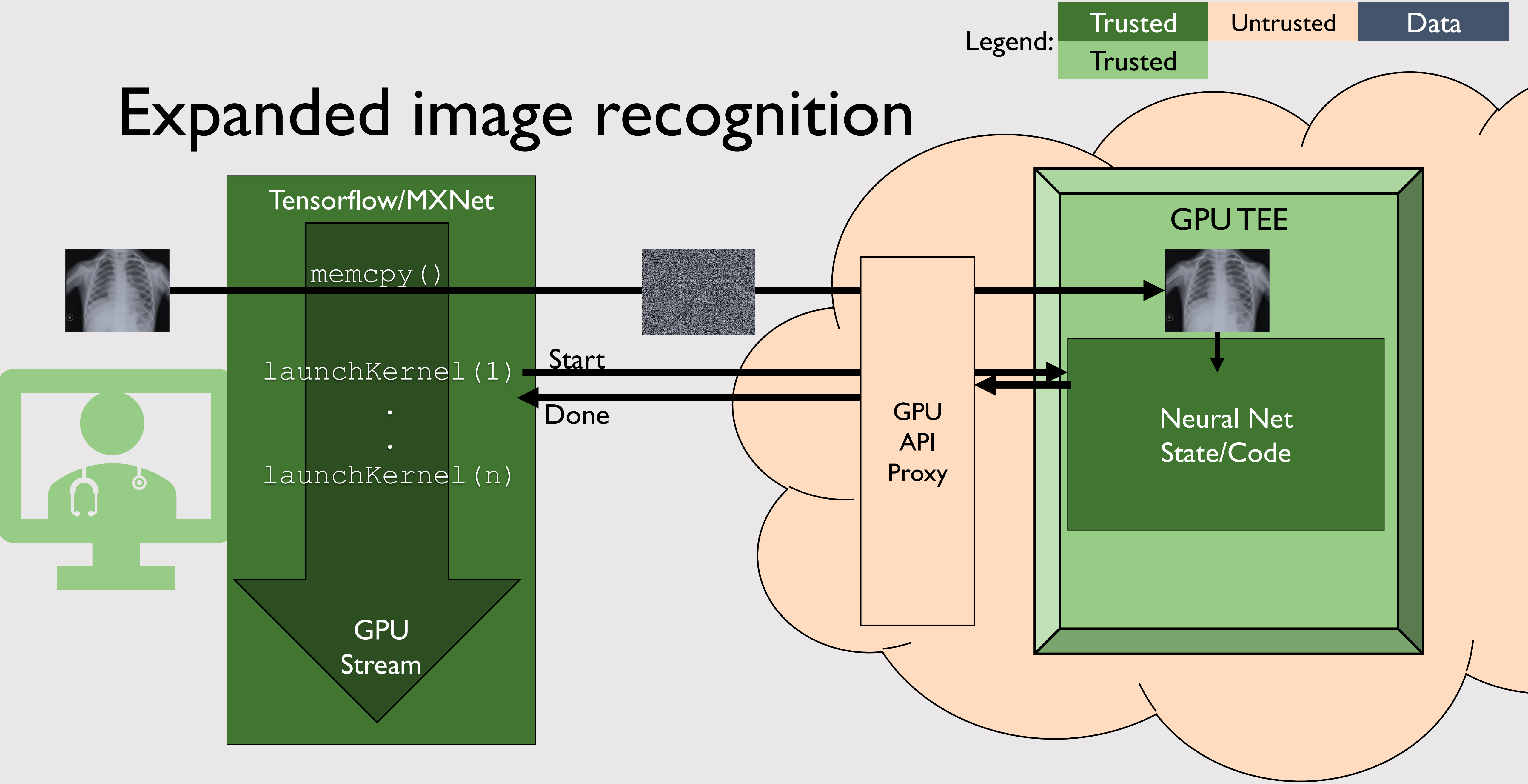
Expanded image recognition



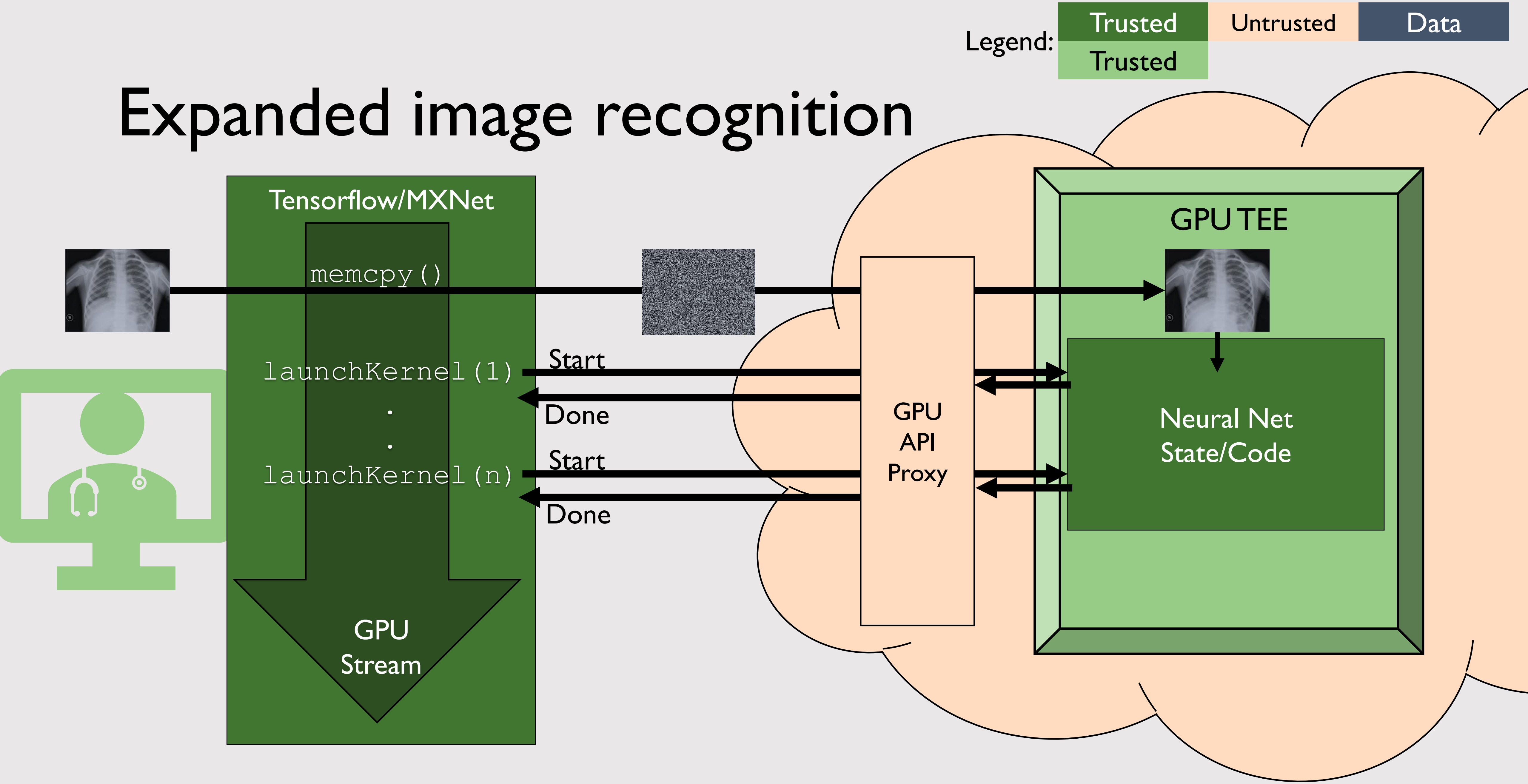
Expanded image recognition



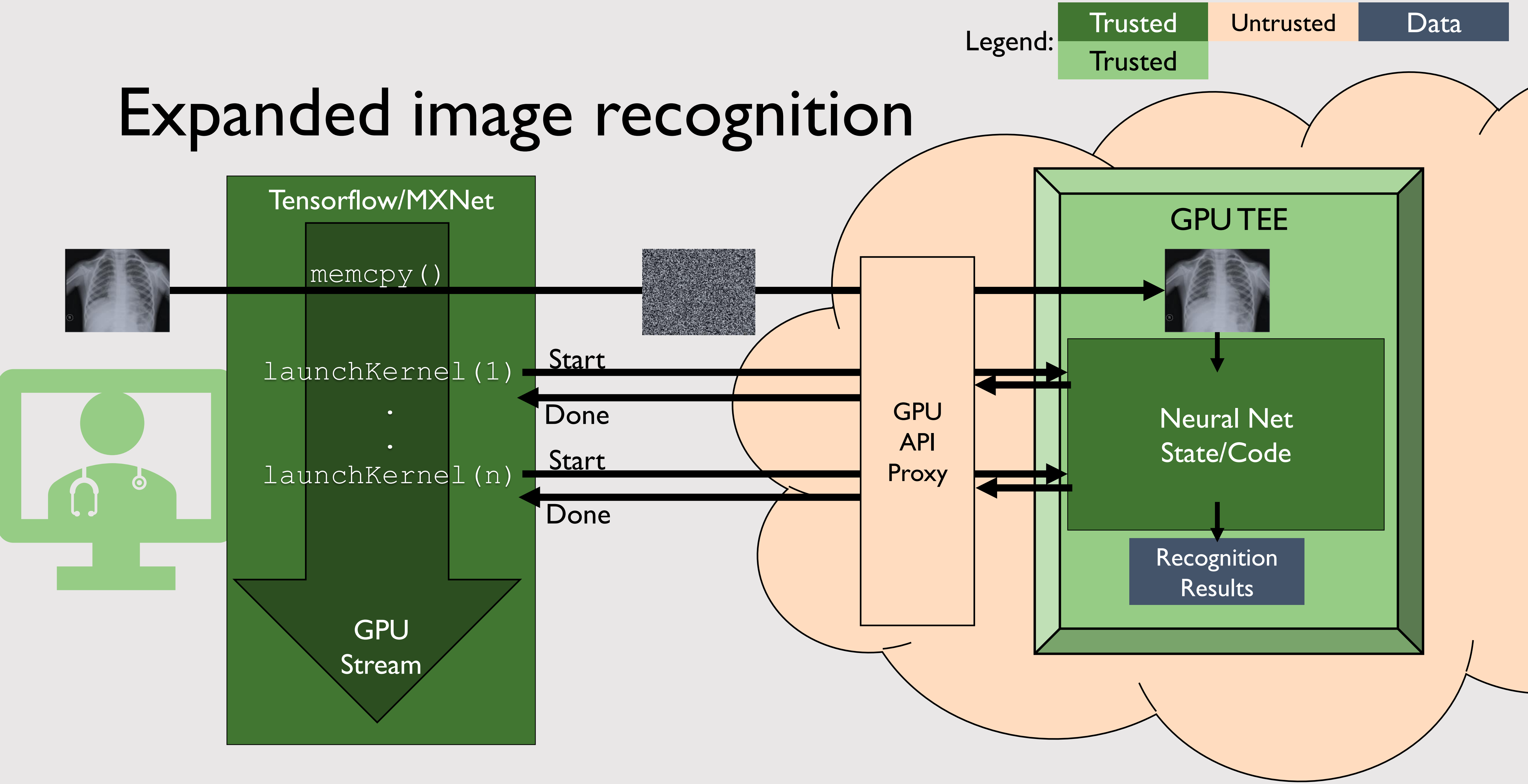
Expanded image recognition



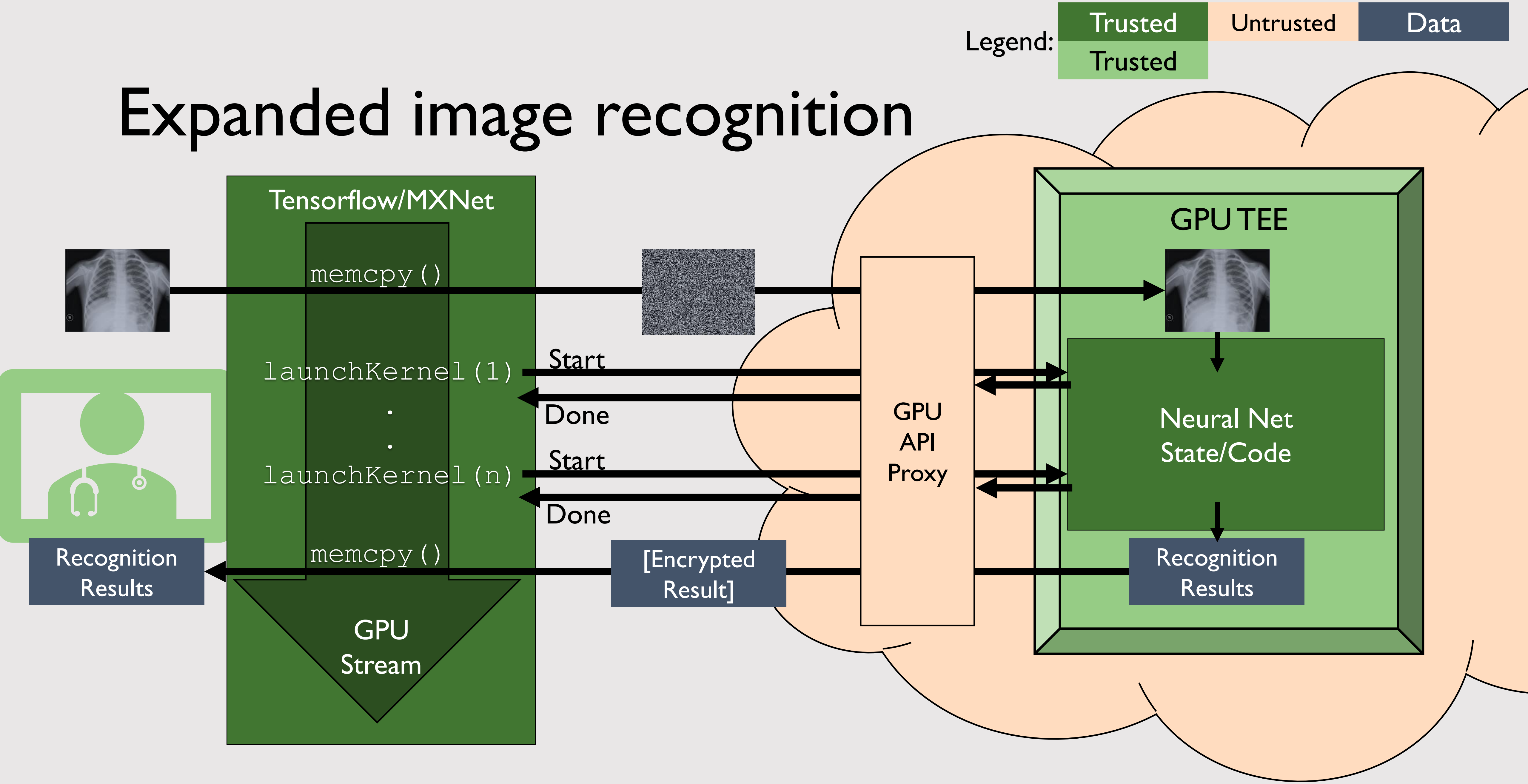
Expanded image recognition



Expanded image recognition

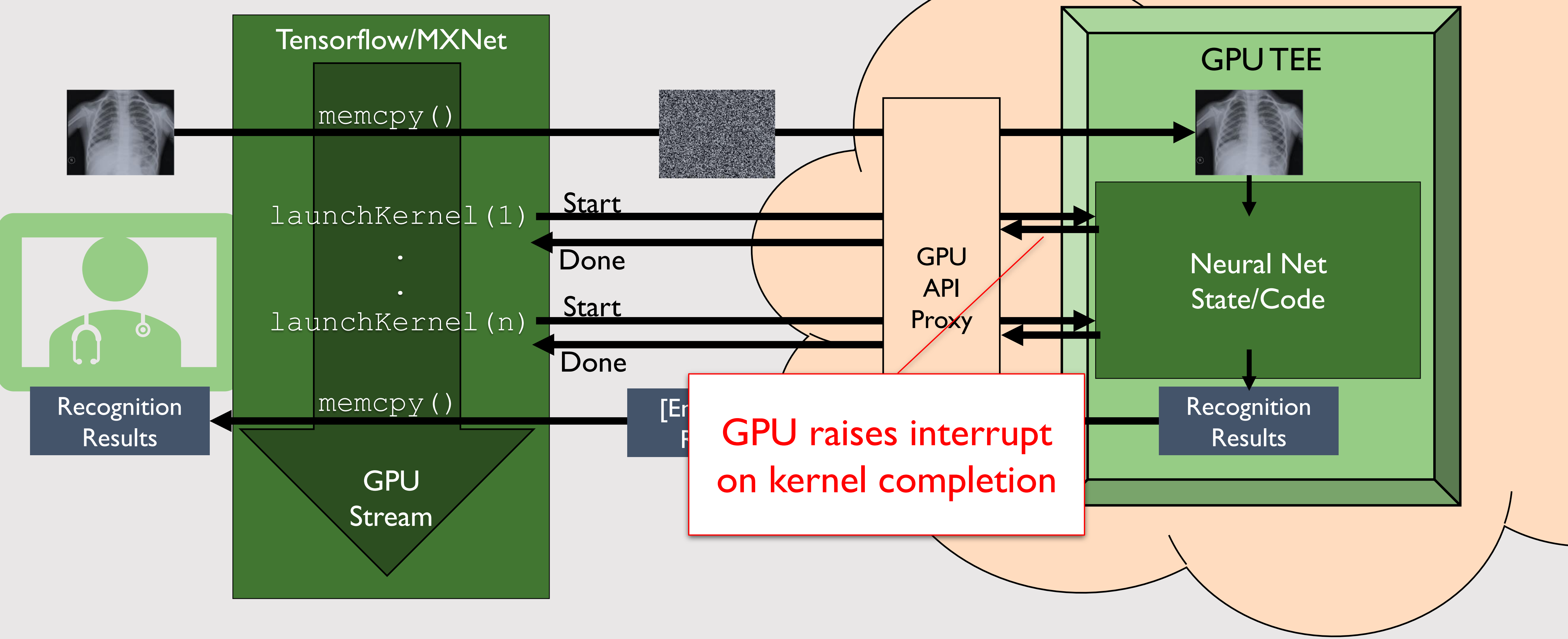


Expanded image recognition

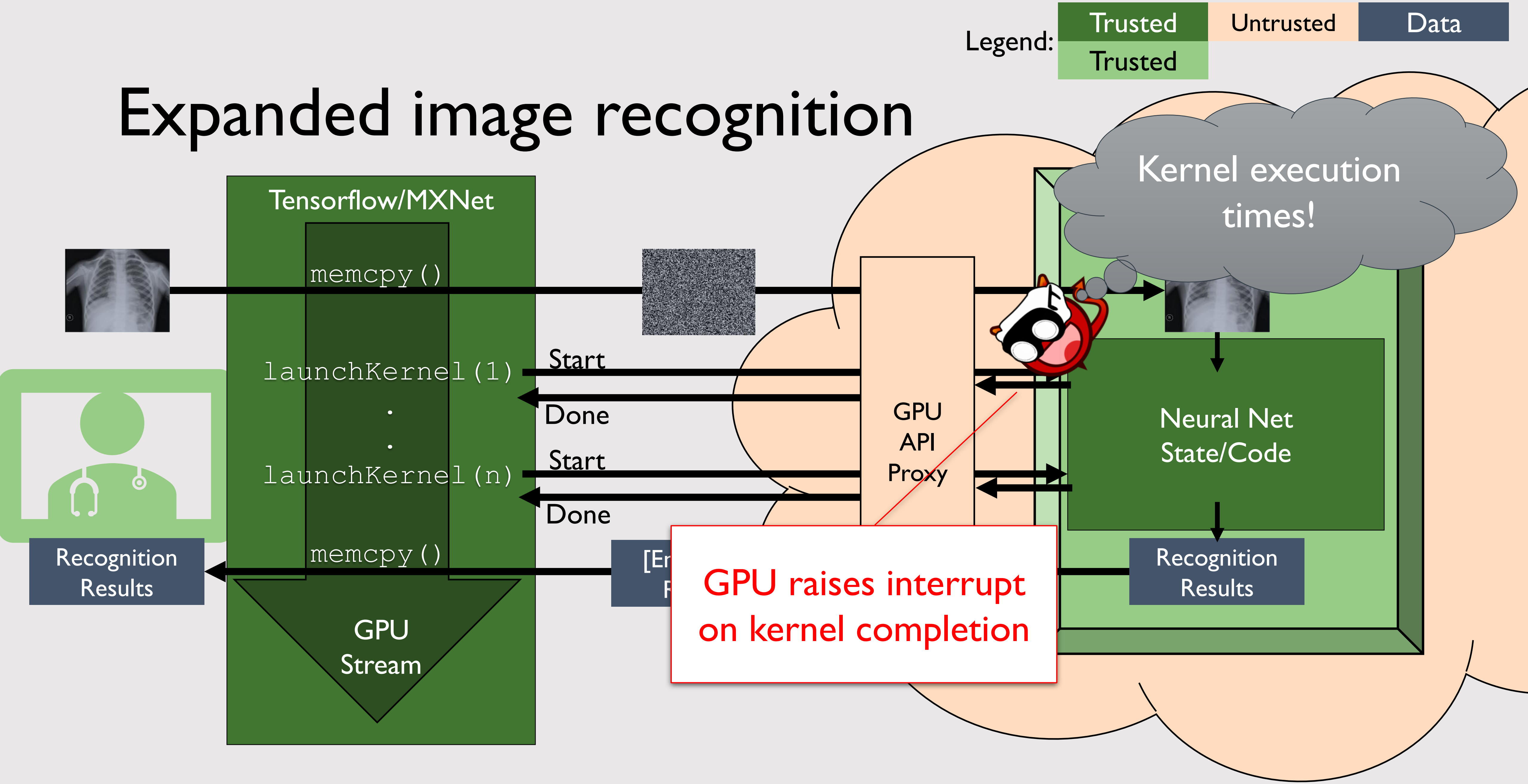


Expanded image recognition

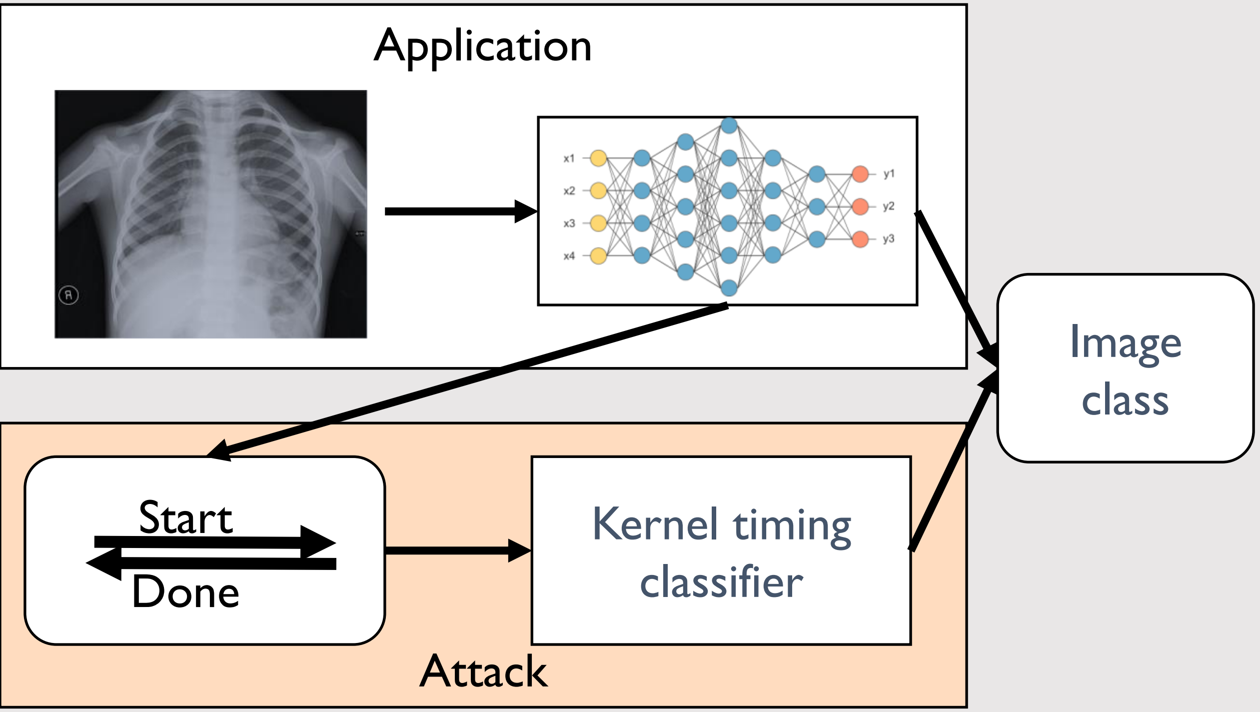
Legend: Trusted | Trusted | Untrusted | Data



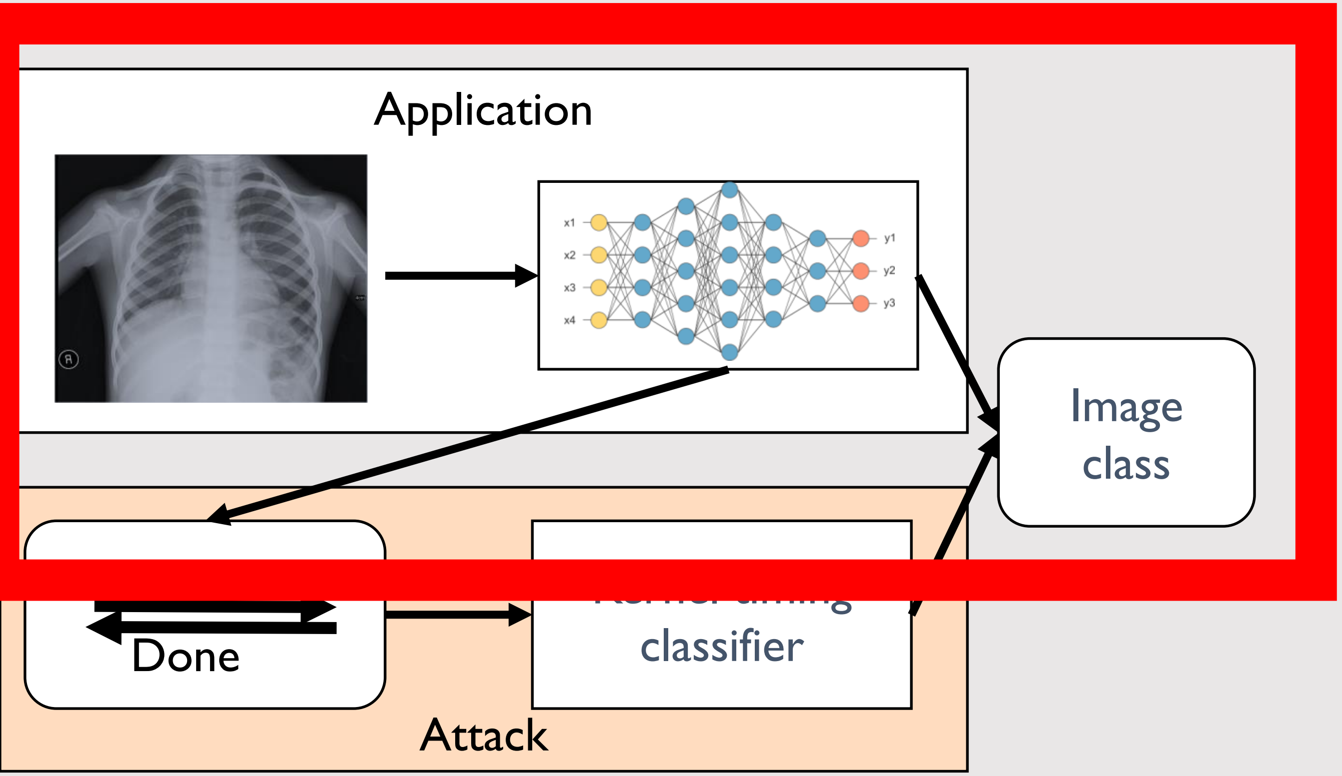
Expanded image recognition



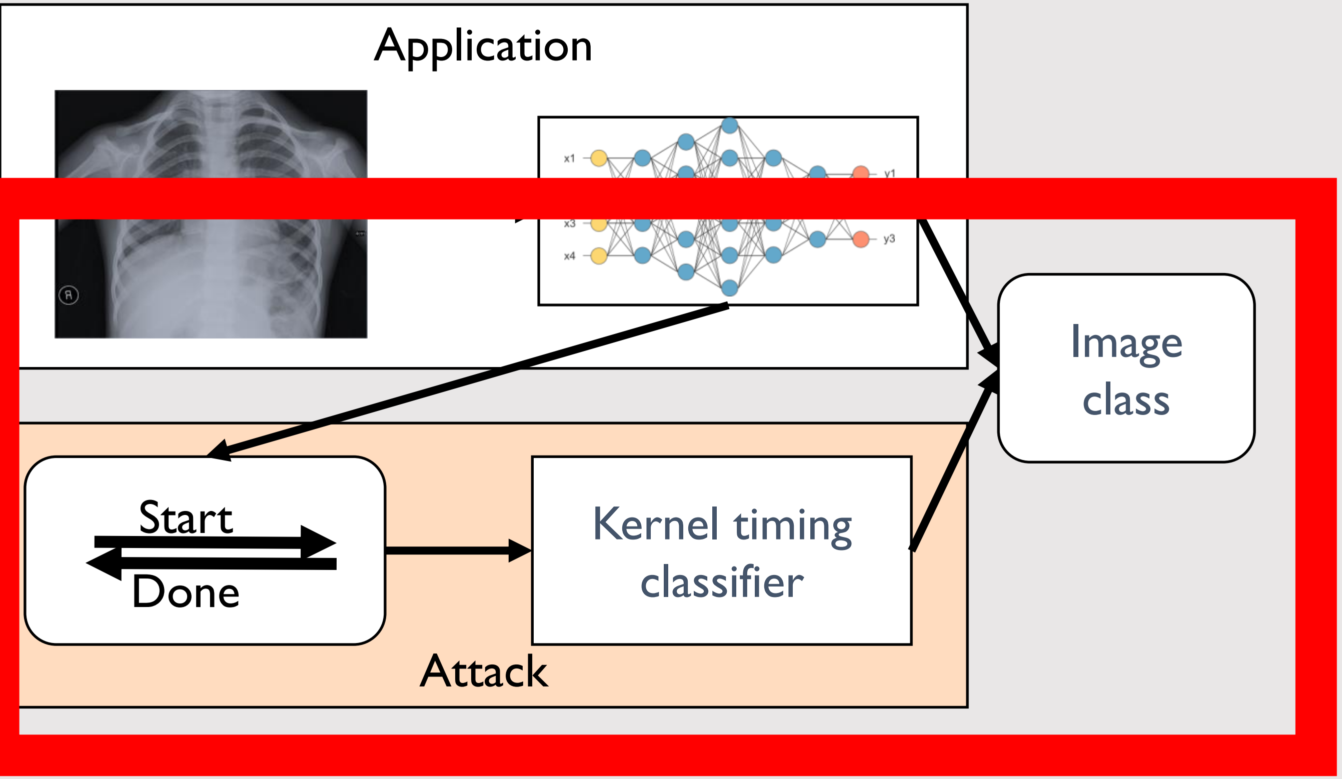
Information gained from kernel execution time



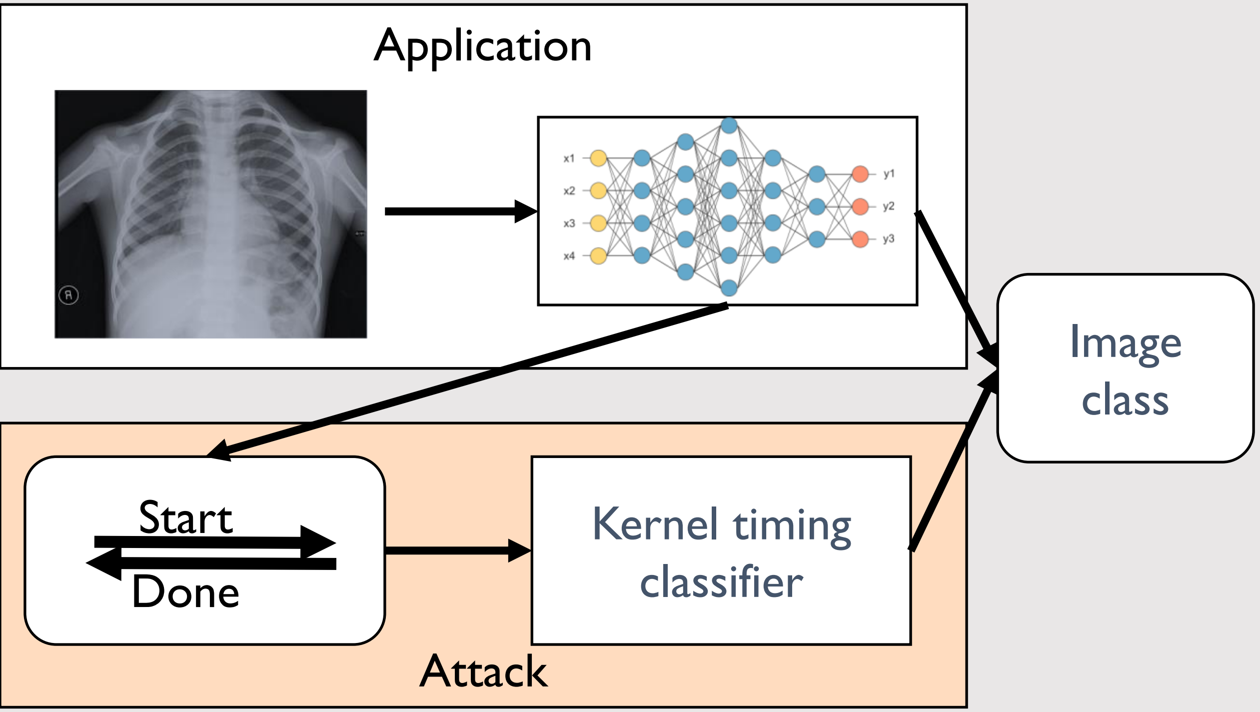
Information gained from kernel execution time



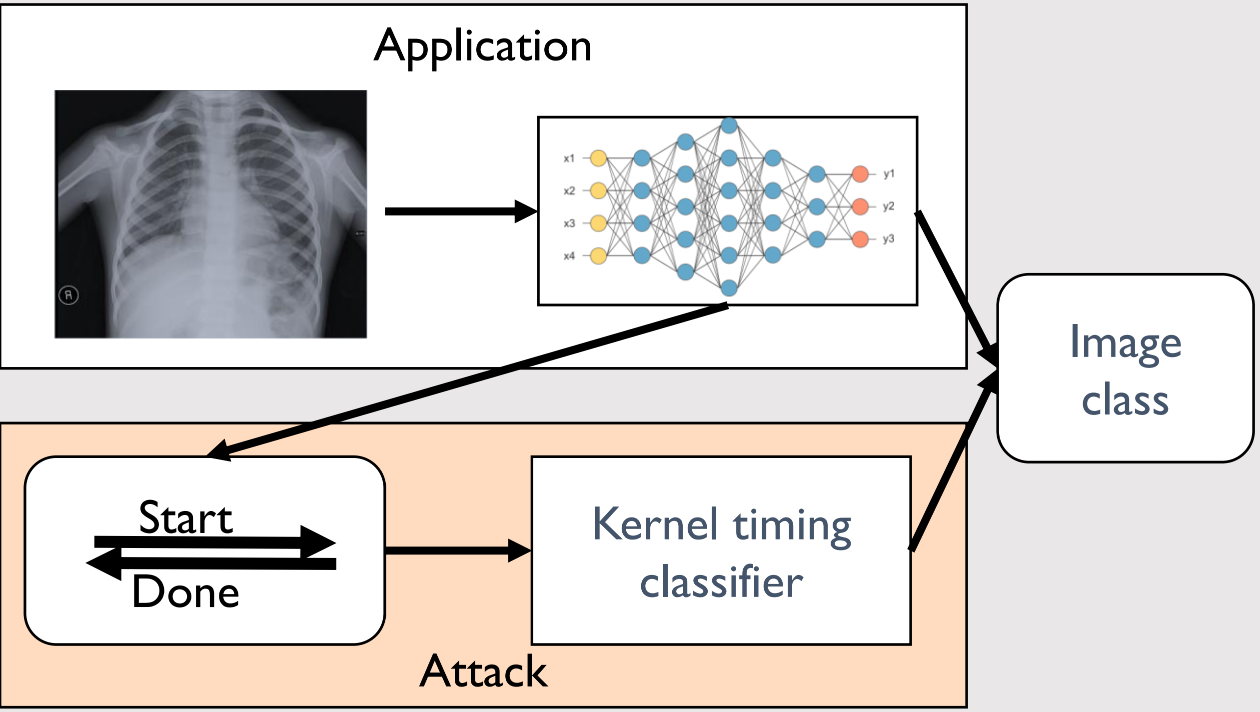
Information gained from kernel execution time

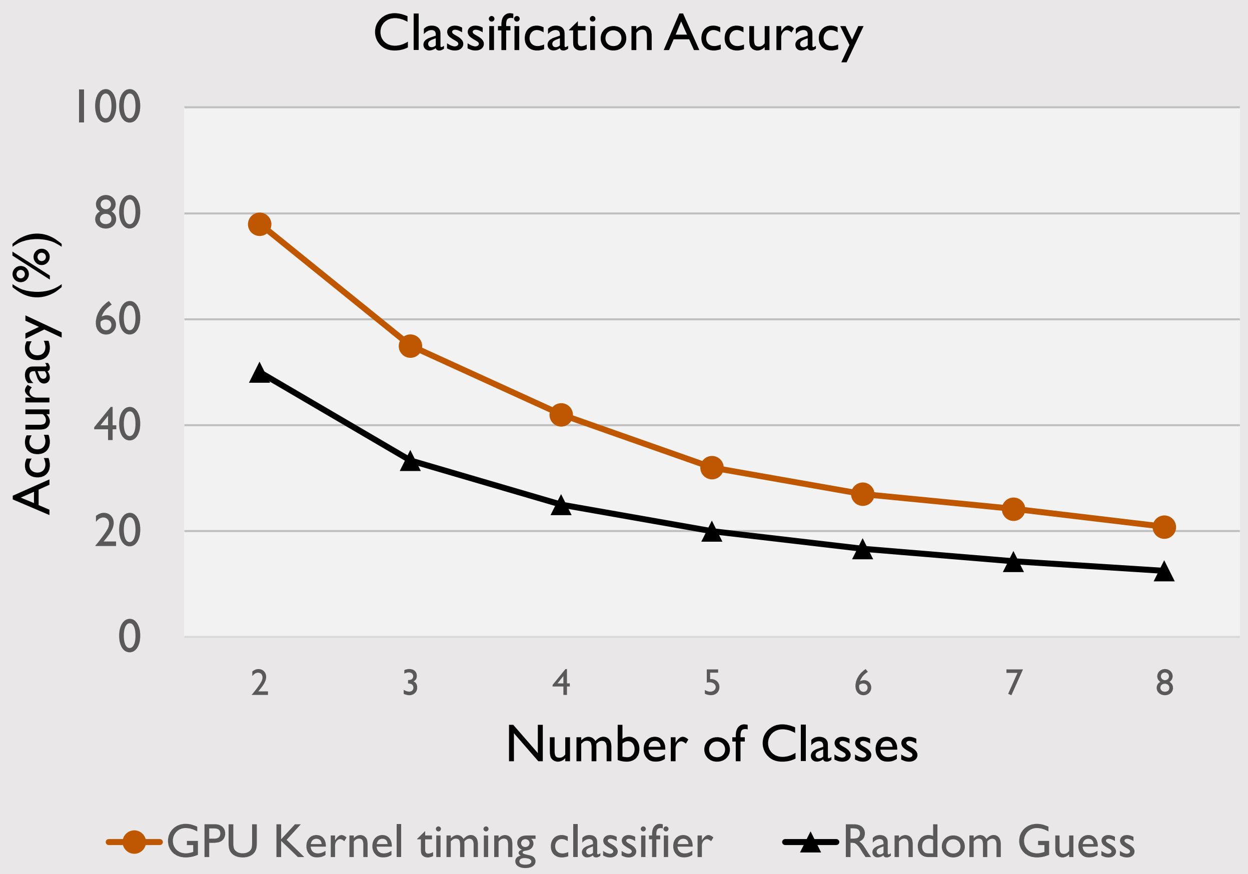


Information gained from kernel execution time

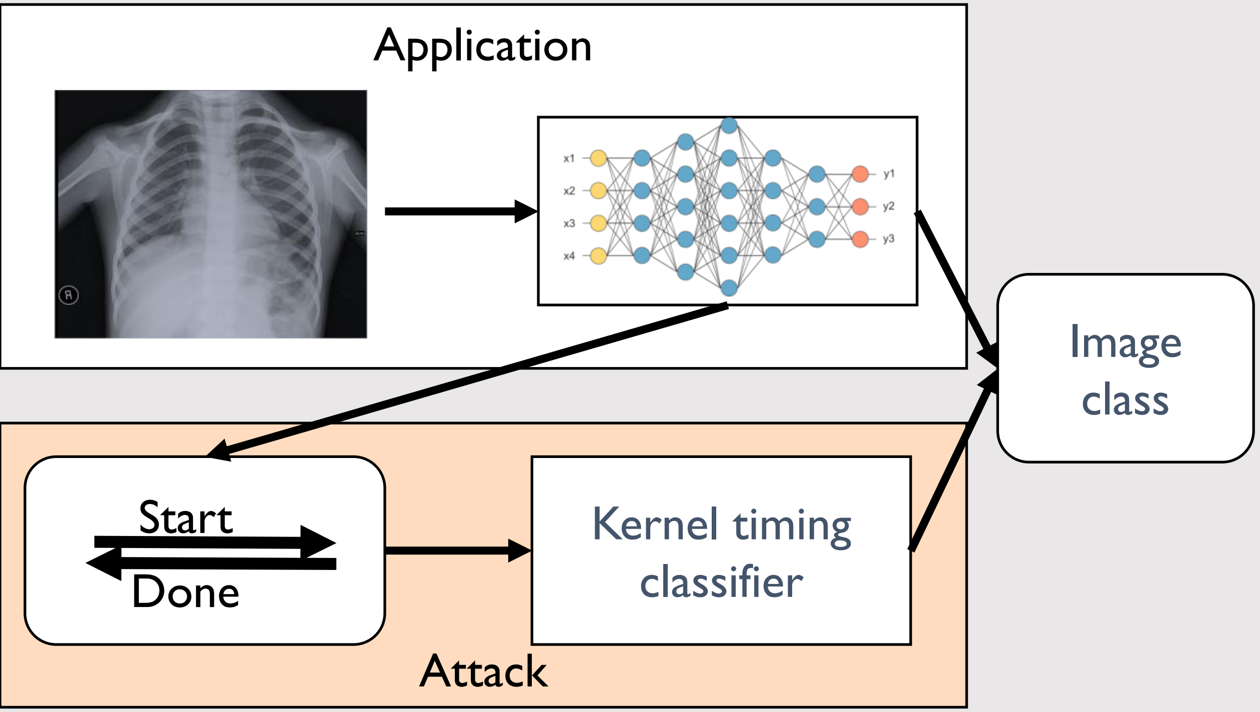


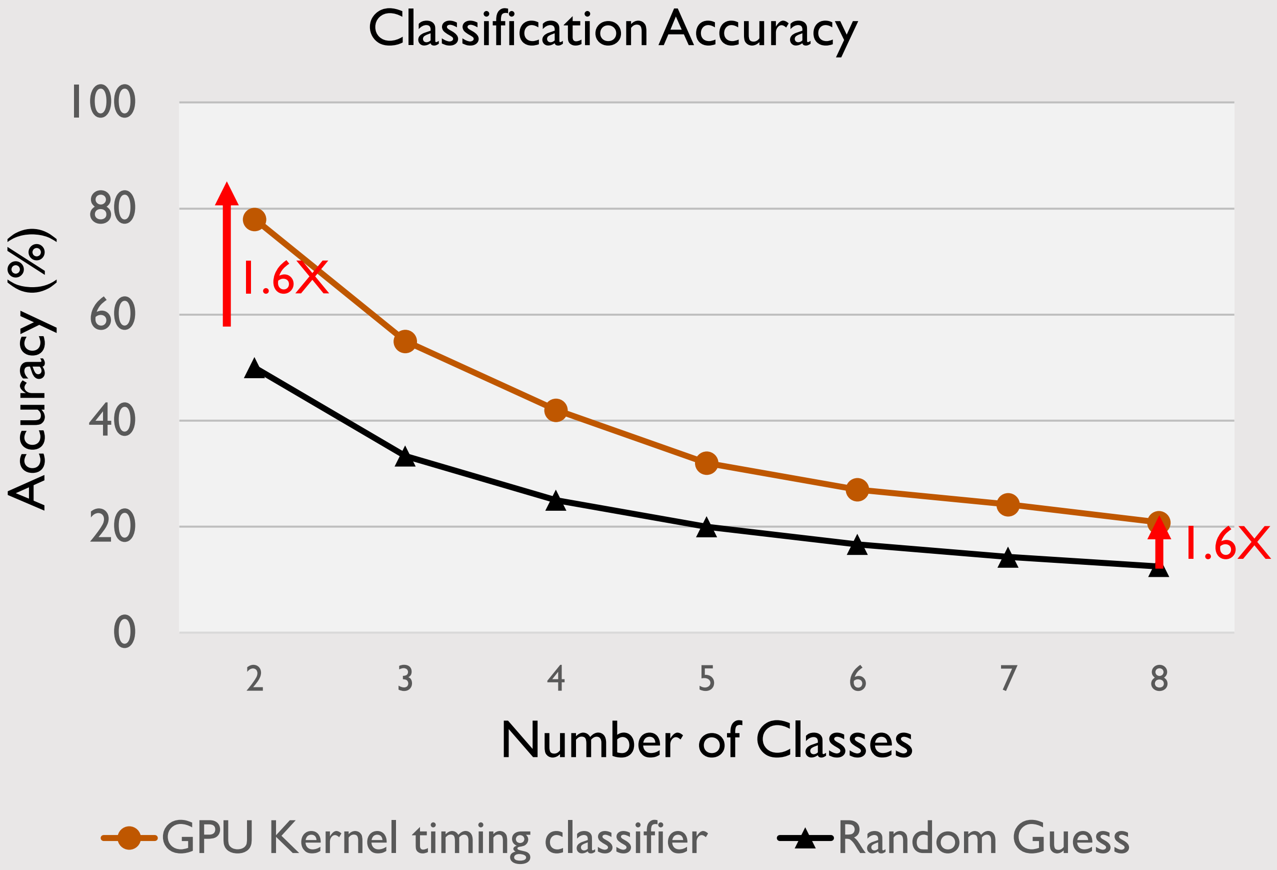
Information gained from kernel execution time





Information gained from kernel execution time





In the rest of the talk we will answer:

- Can information be extracted from GPU communication patterns?

  Yes, we demonstrate a communication timing attack

- How does Telekine remove that information?

  Replace GPU streams with new data-oblivious streams

- What are Telekine's overheads?

  Overheads are reasonable: ~20% for neural network training

In the rest of the talk we will answer:

- Can information be extracted from GPU communication patterns?

  Yes, we demonstrate a communication timing attack

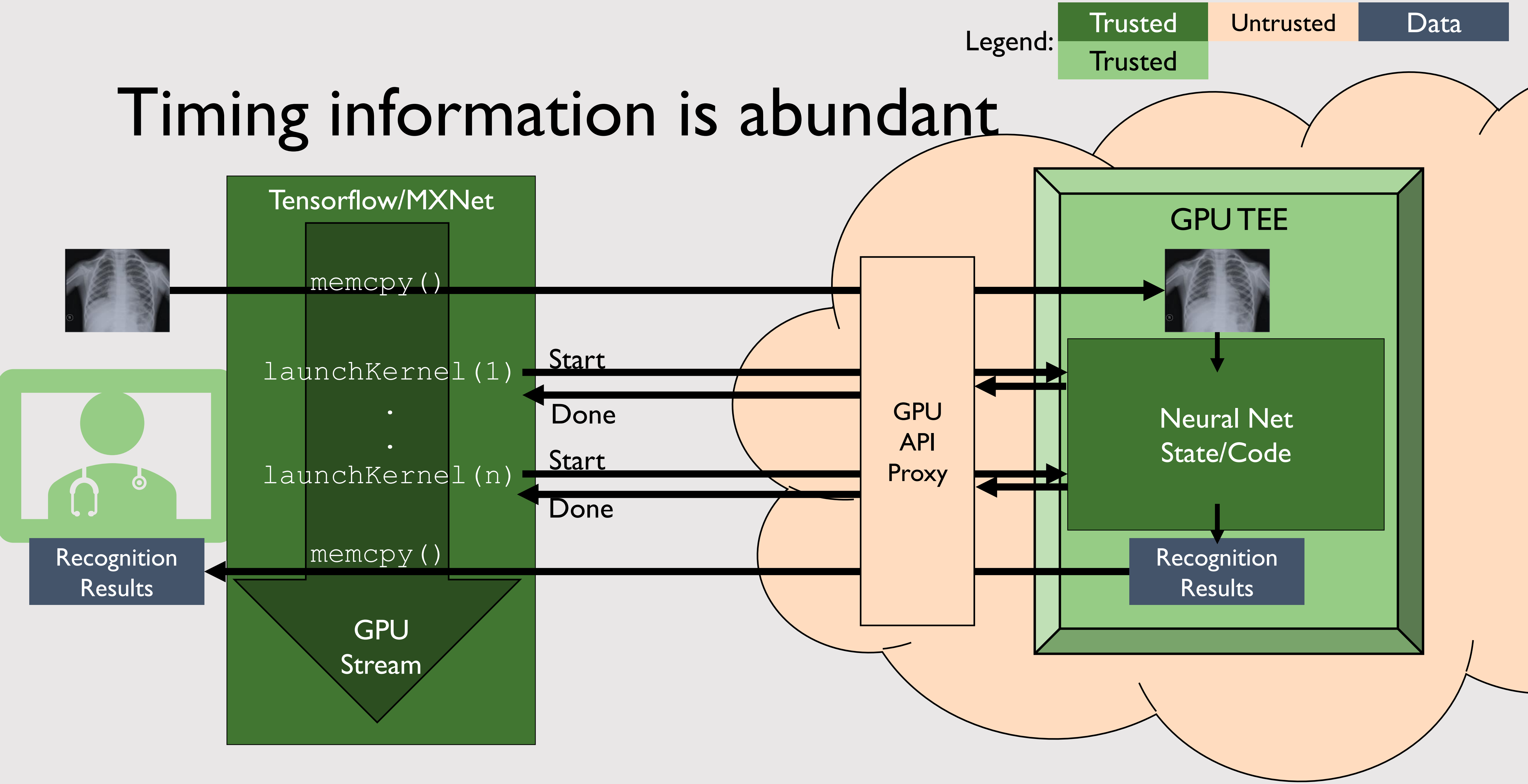
- How does Telekine remove that information?

  Replace GPU streams with new data-oblivious streams

- What are Telekine's overheads?

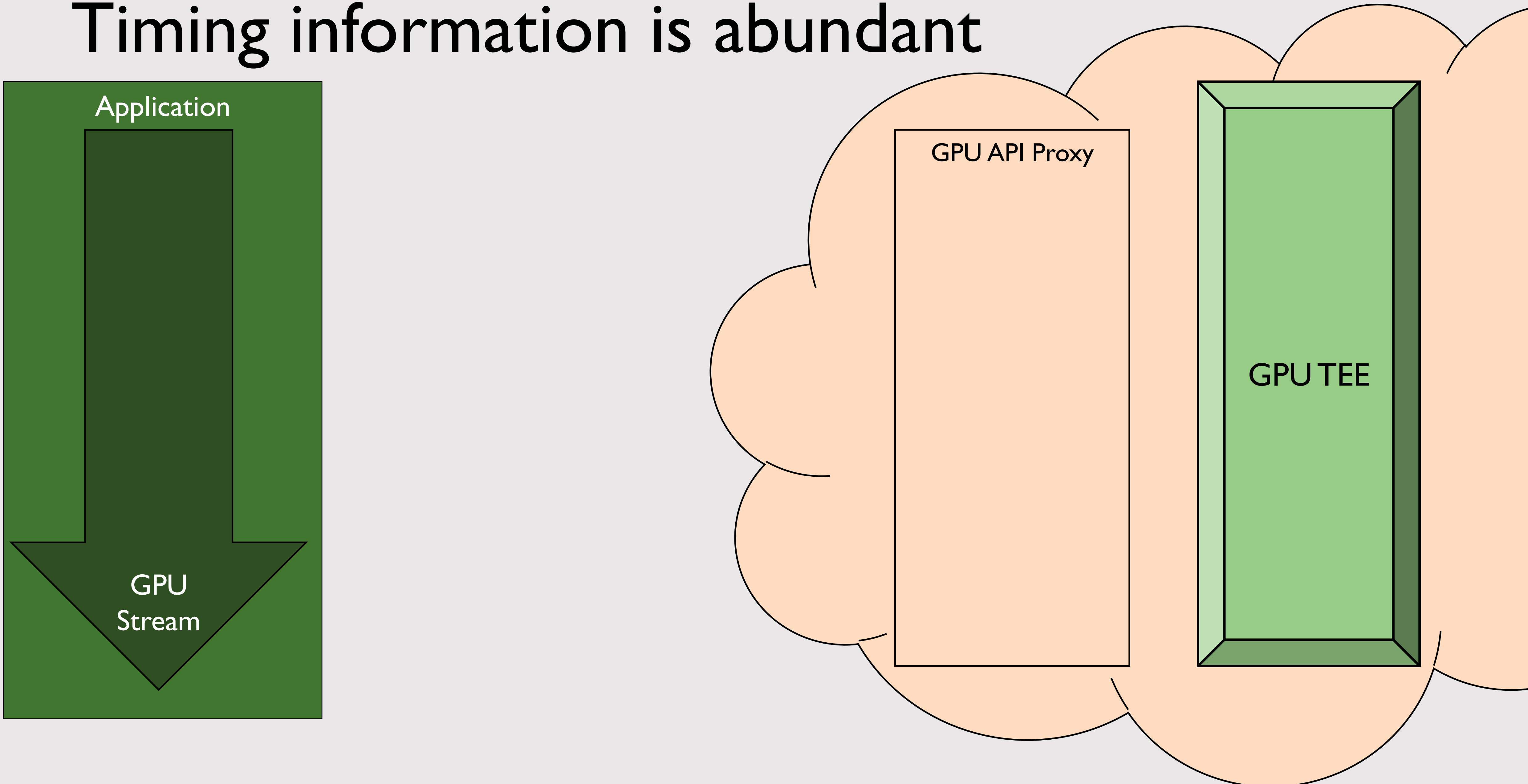
  Overheads are reasonable: ~20% for neural network training

Timing information is abundant



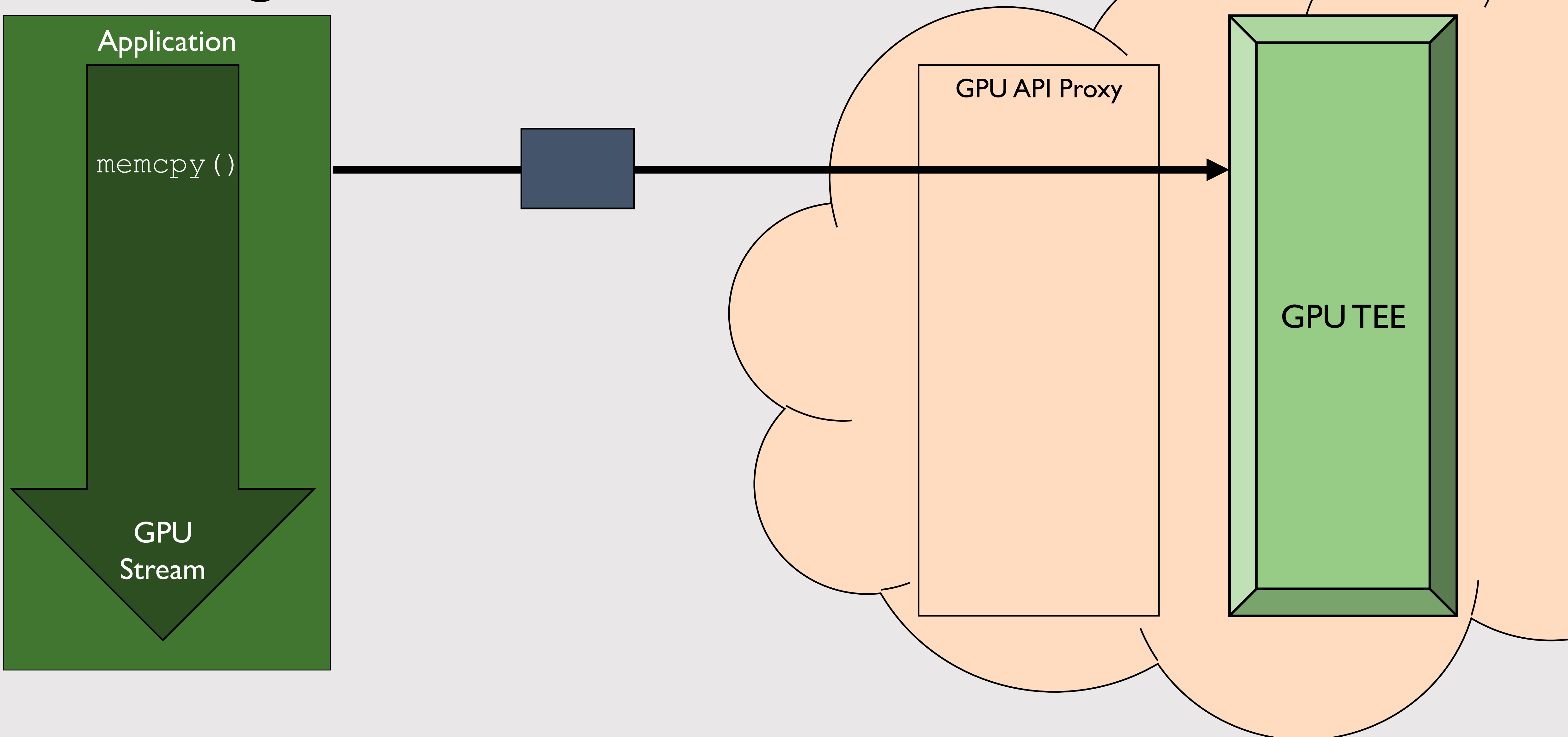
Legend: Trusted | Untrusted | Data | Trusted

Timing information is abundant

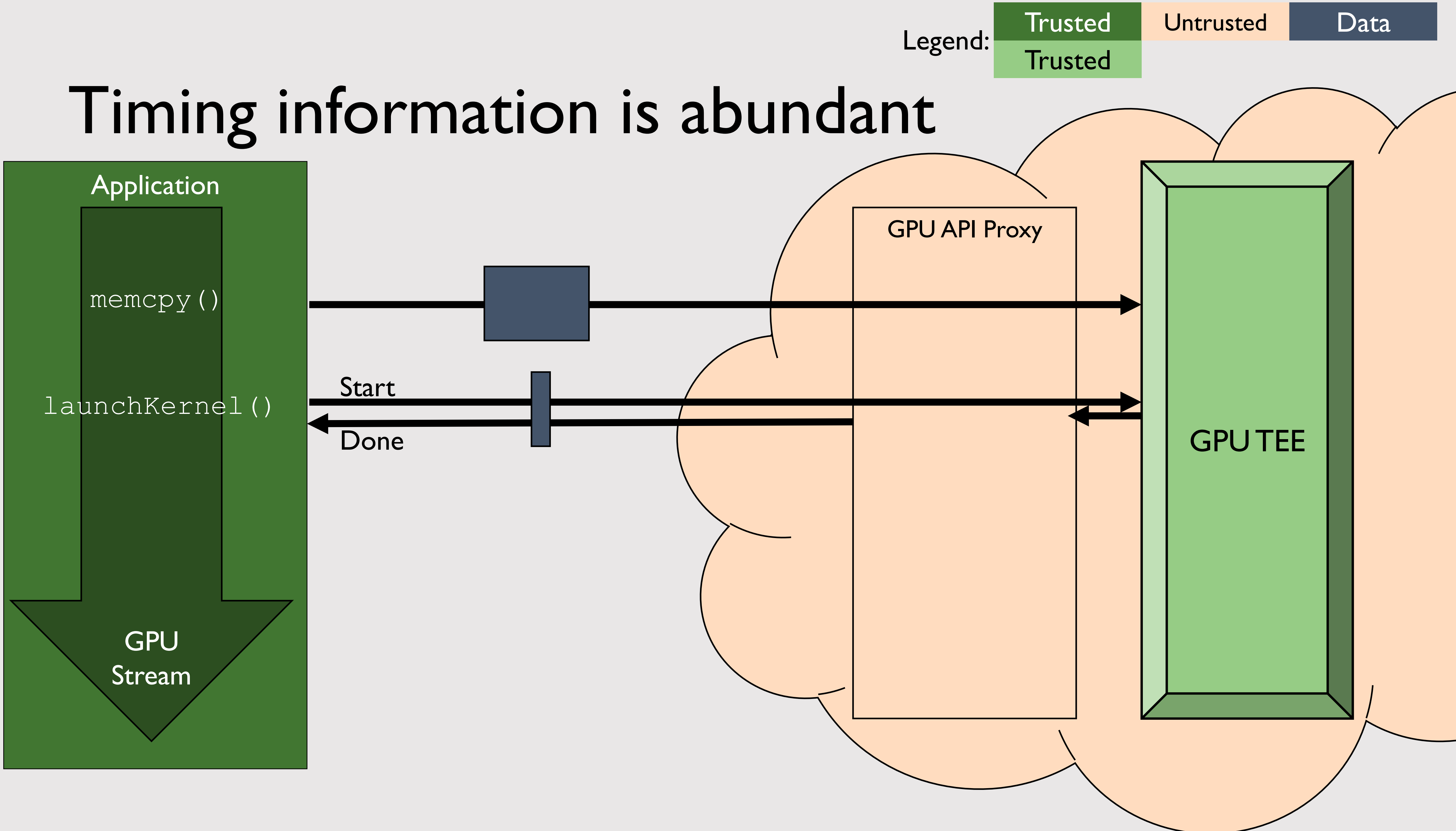


Legend: Trusted | Untrusted | Data | Trusted

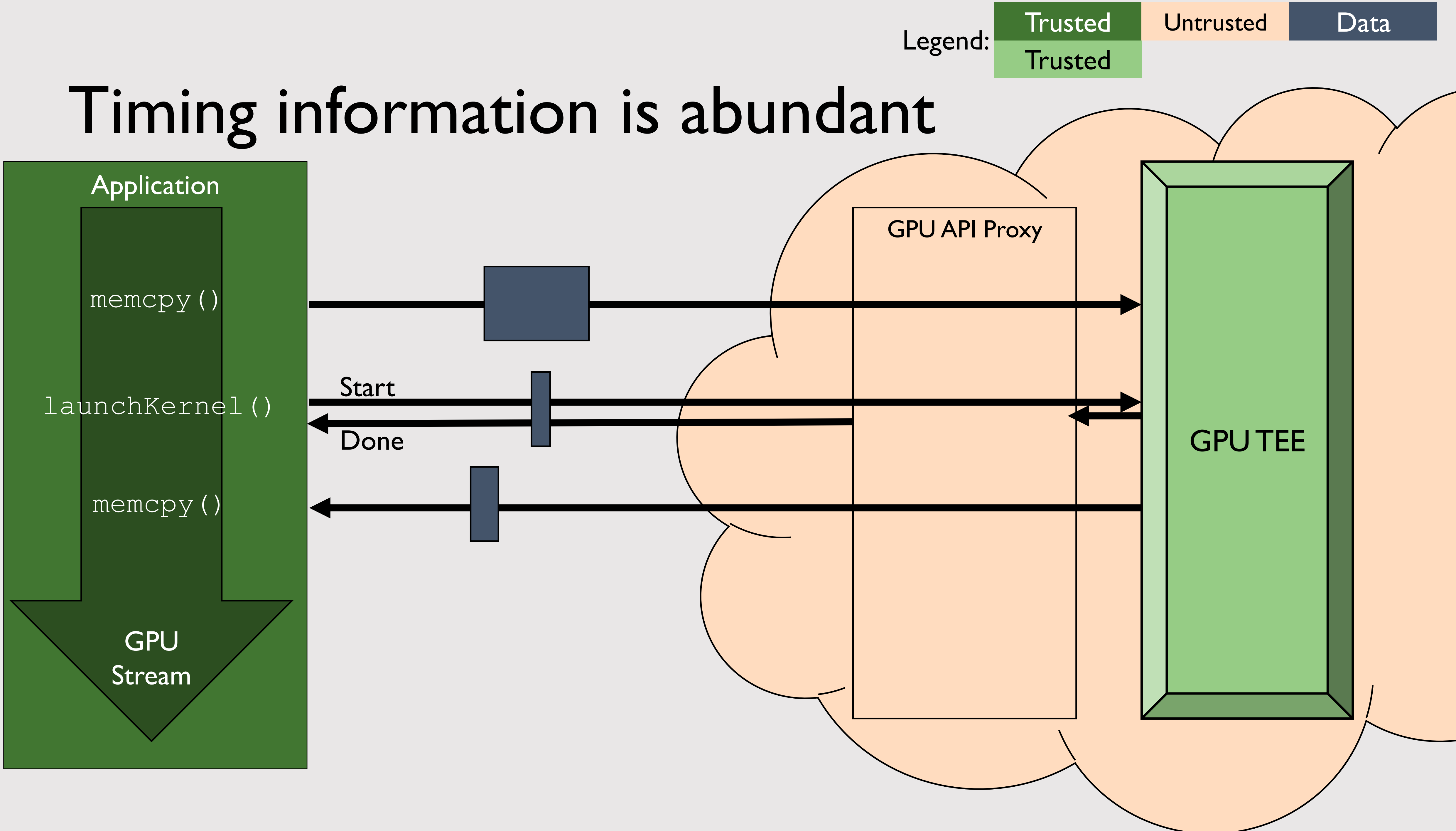
Timing information is abundant



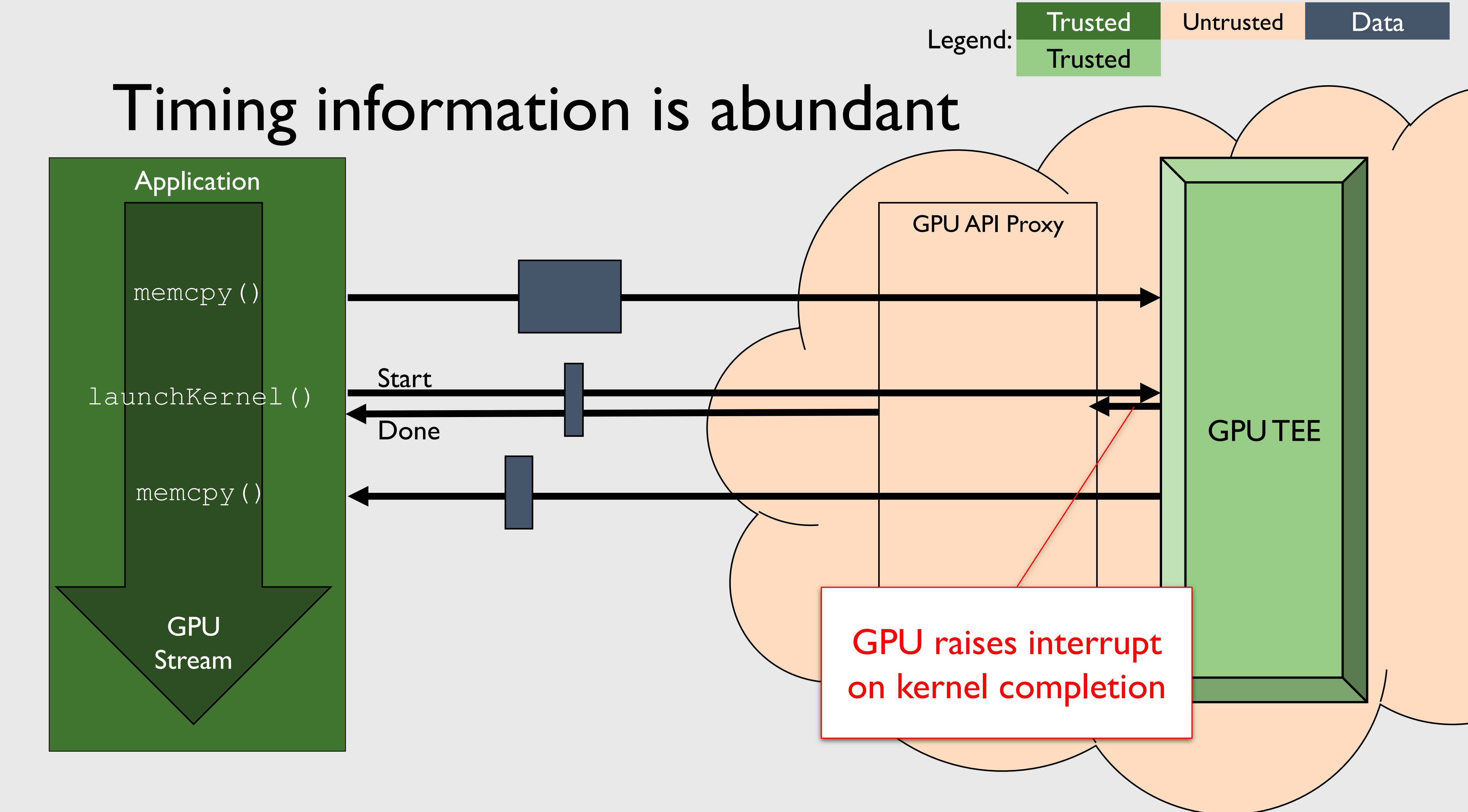
Timing information is abundant



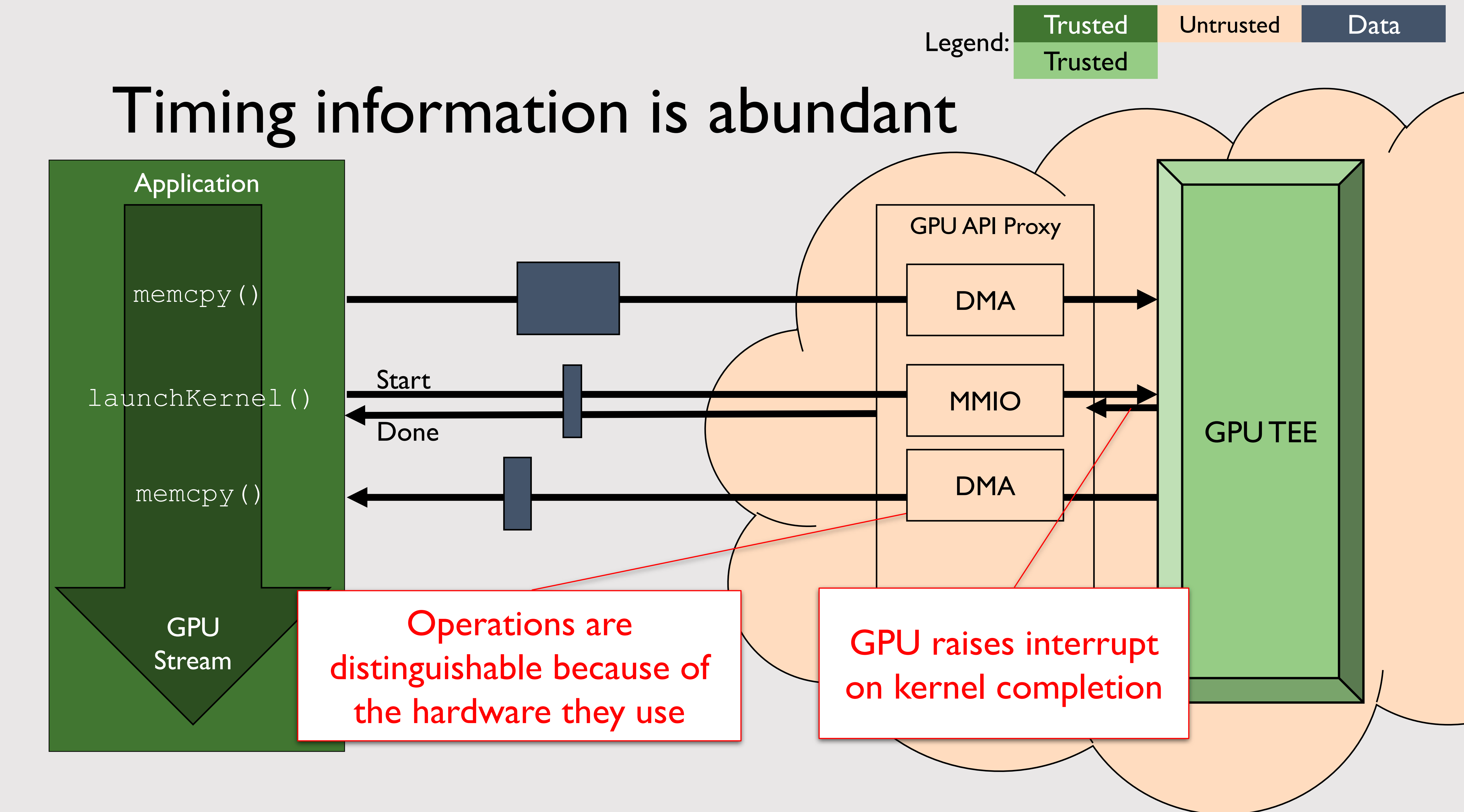
Timing information is abundant



Timing information is abundant

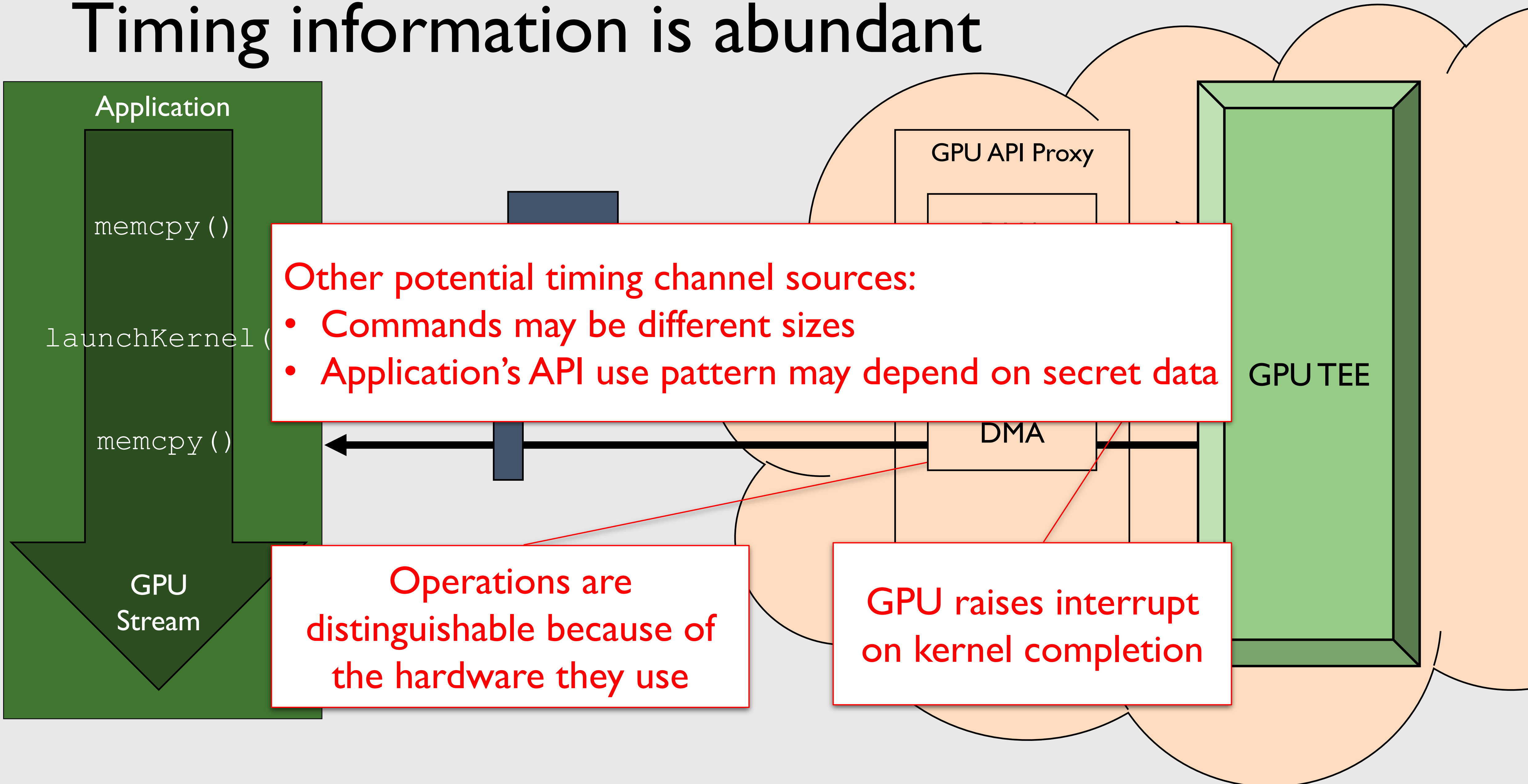


Timing information is abundant



Legend: Trusted | Untrusted | Data | Trusted

Timing information is abundant



Other potential timing channel sources:

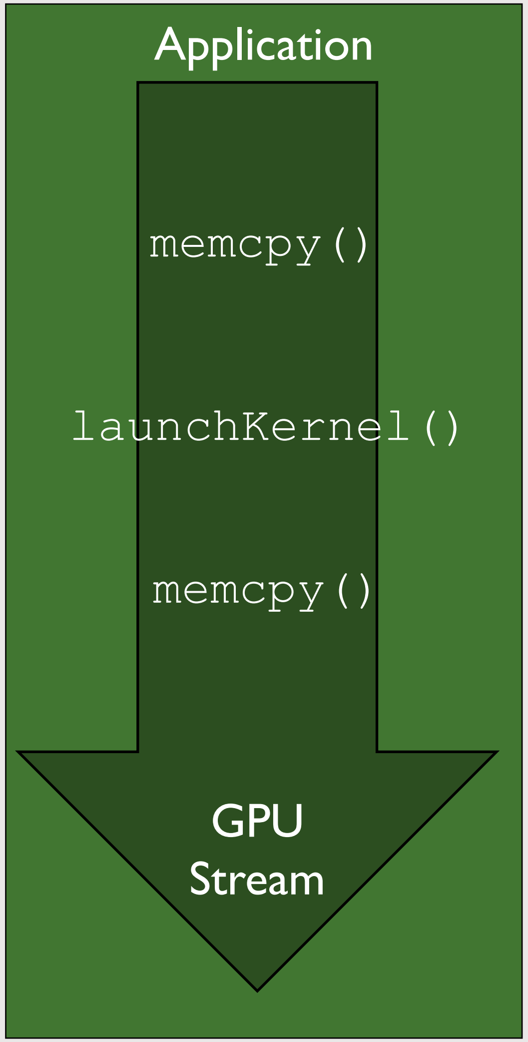
- Commands may be different sizes
- Application’s API use pattern may depend on secret data

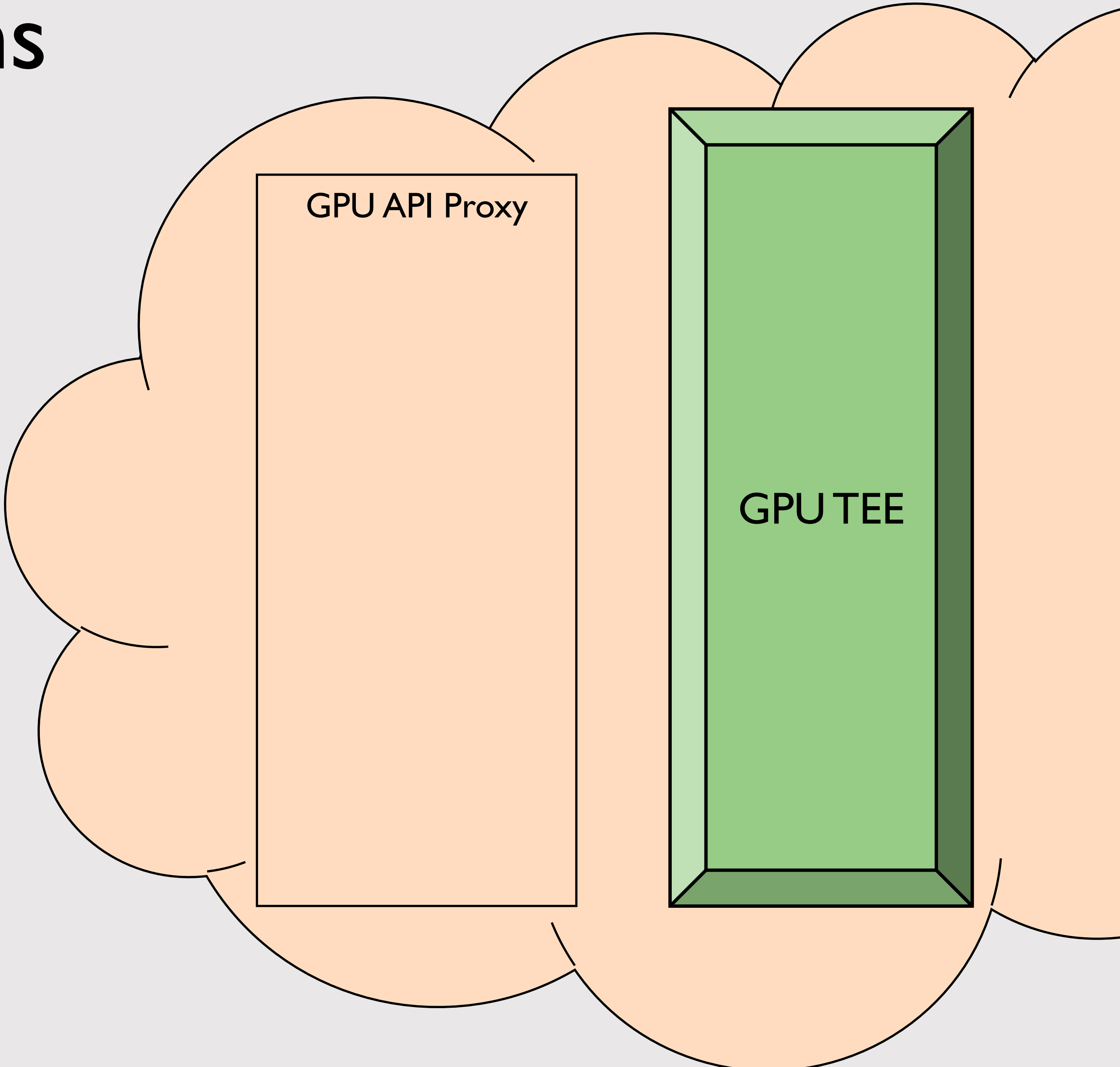
Operations are distinguishable because of the hardware they use

GPU raises interrupt on kernel completion

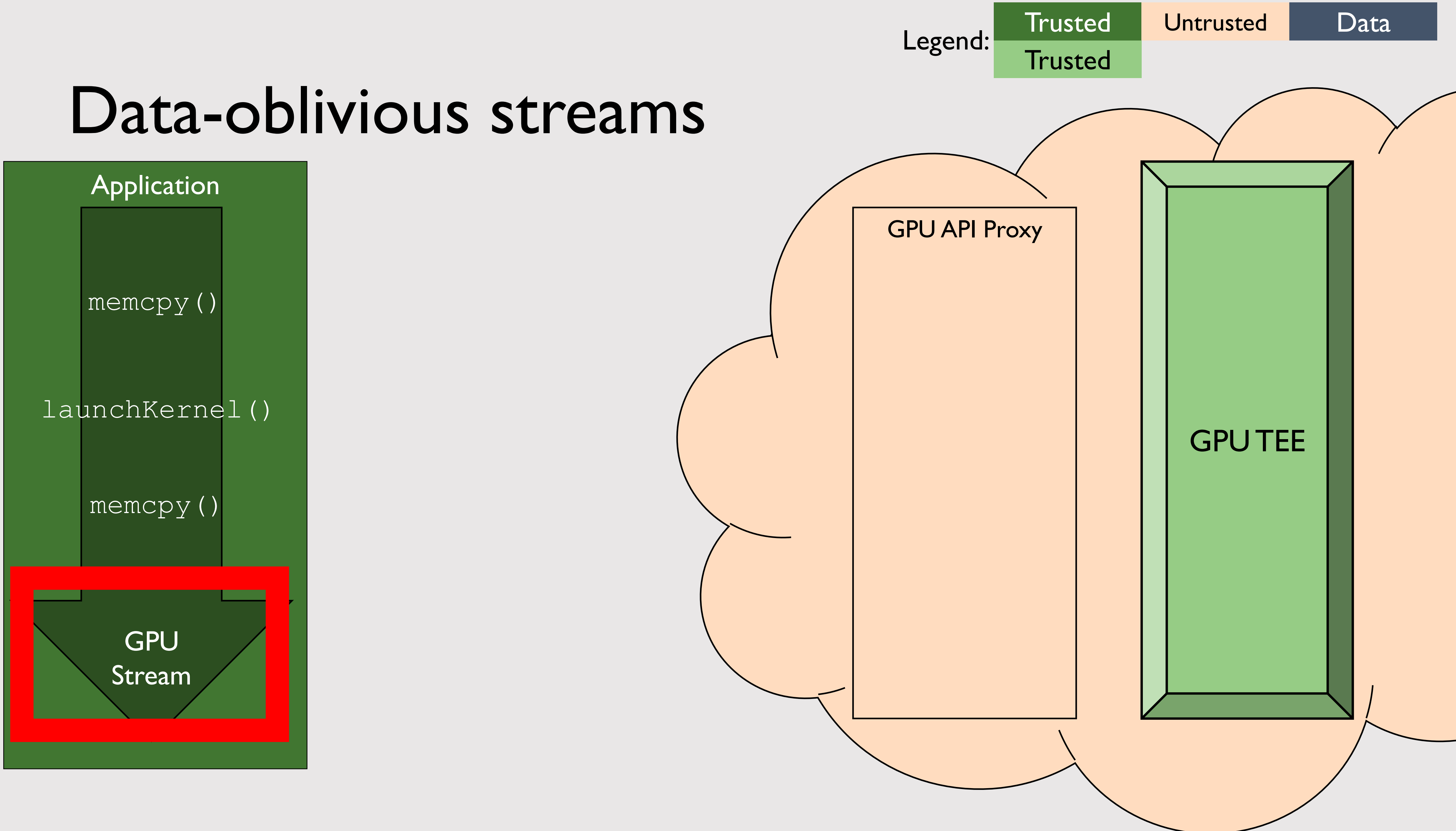
Legend: Trusted | Trusted | Untrusted | Data

Data-oblivious streams



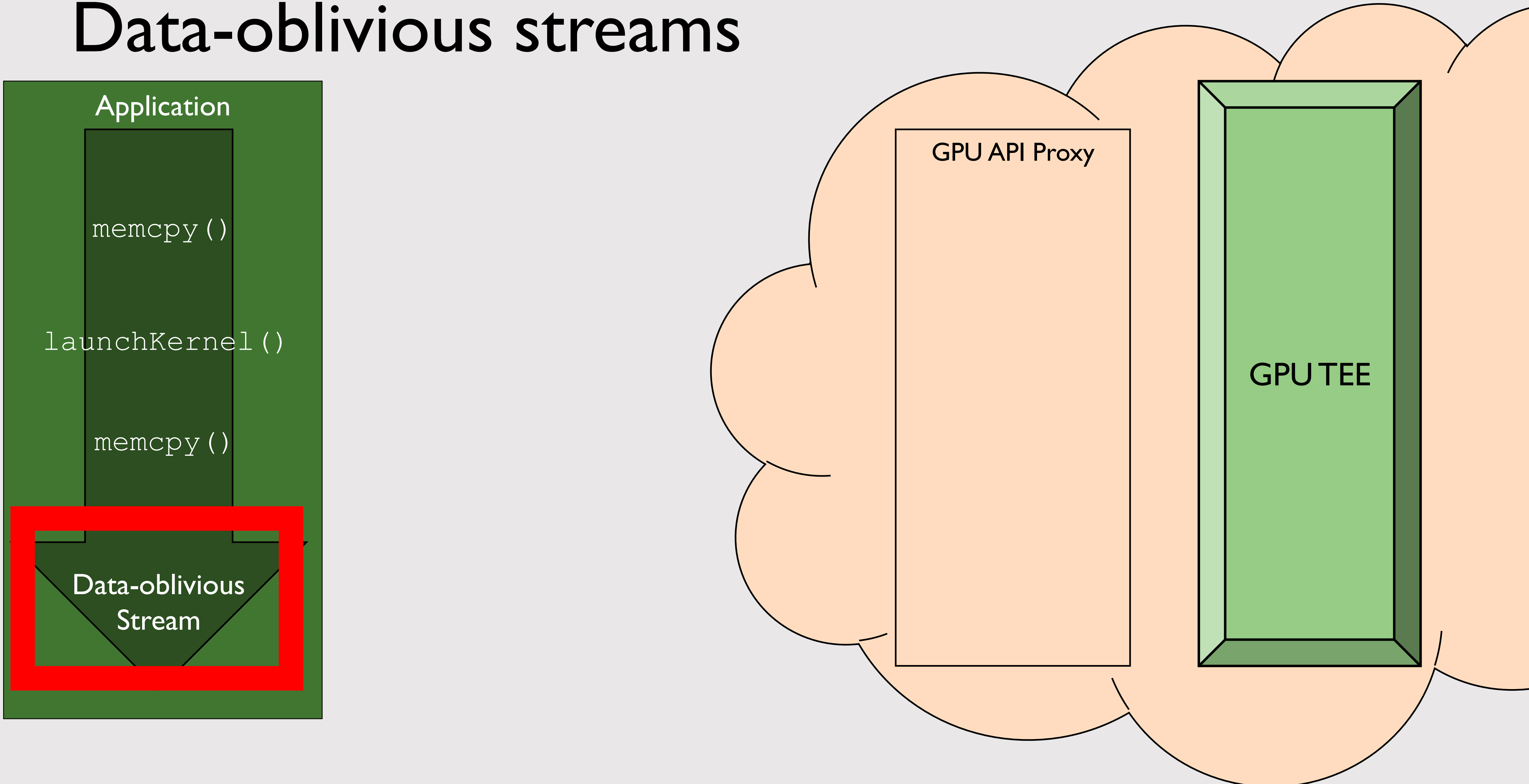


Data-oblivious streams

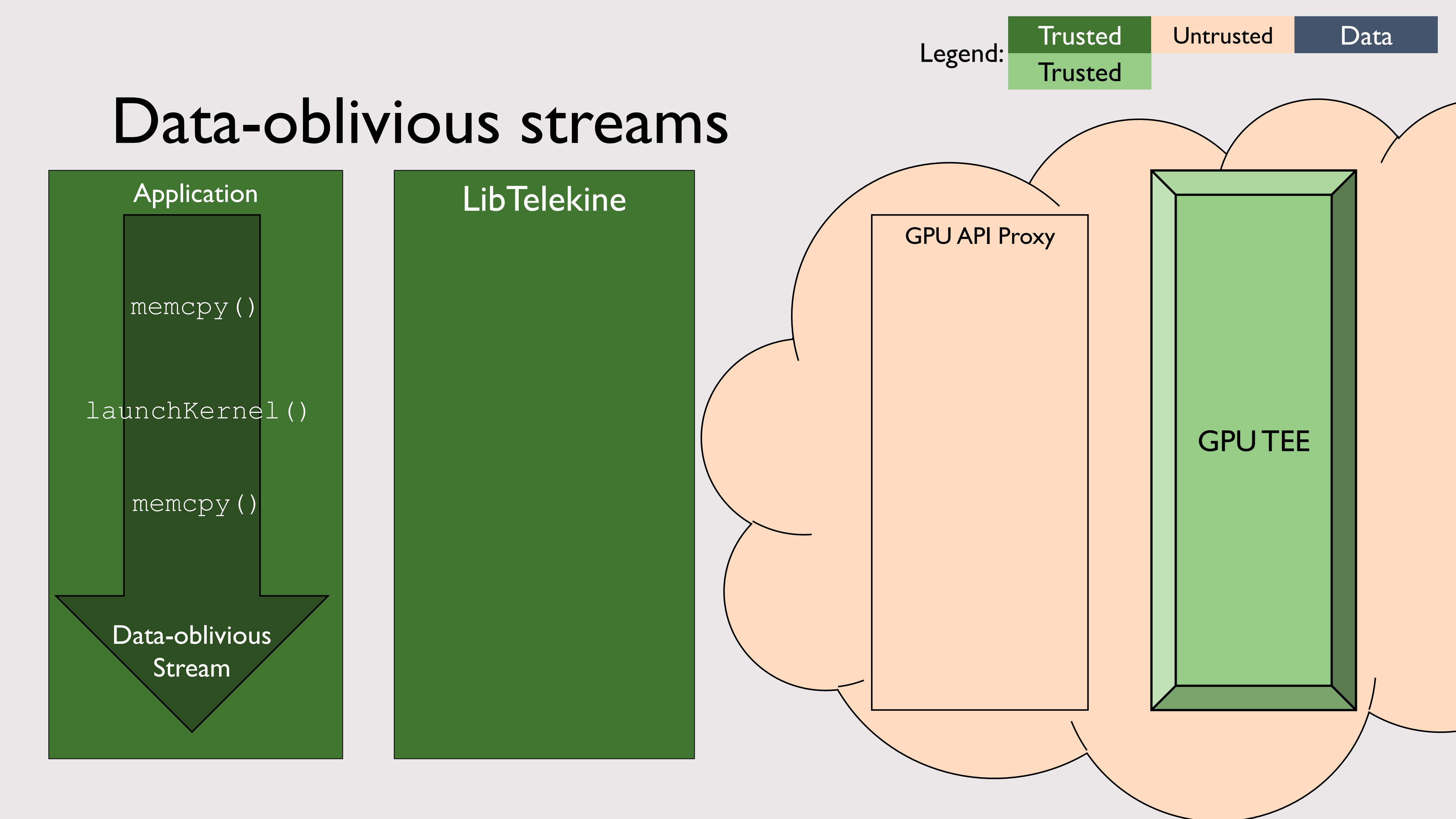


Data-oblivious streams

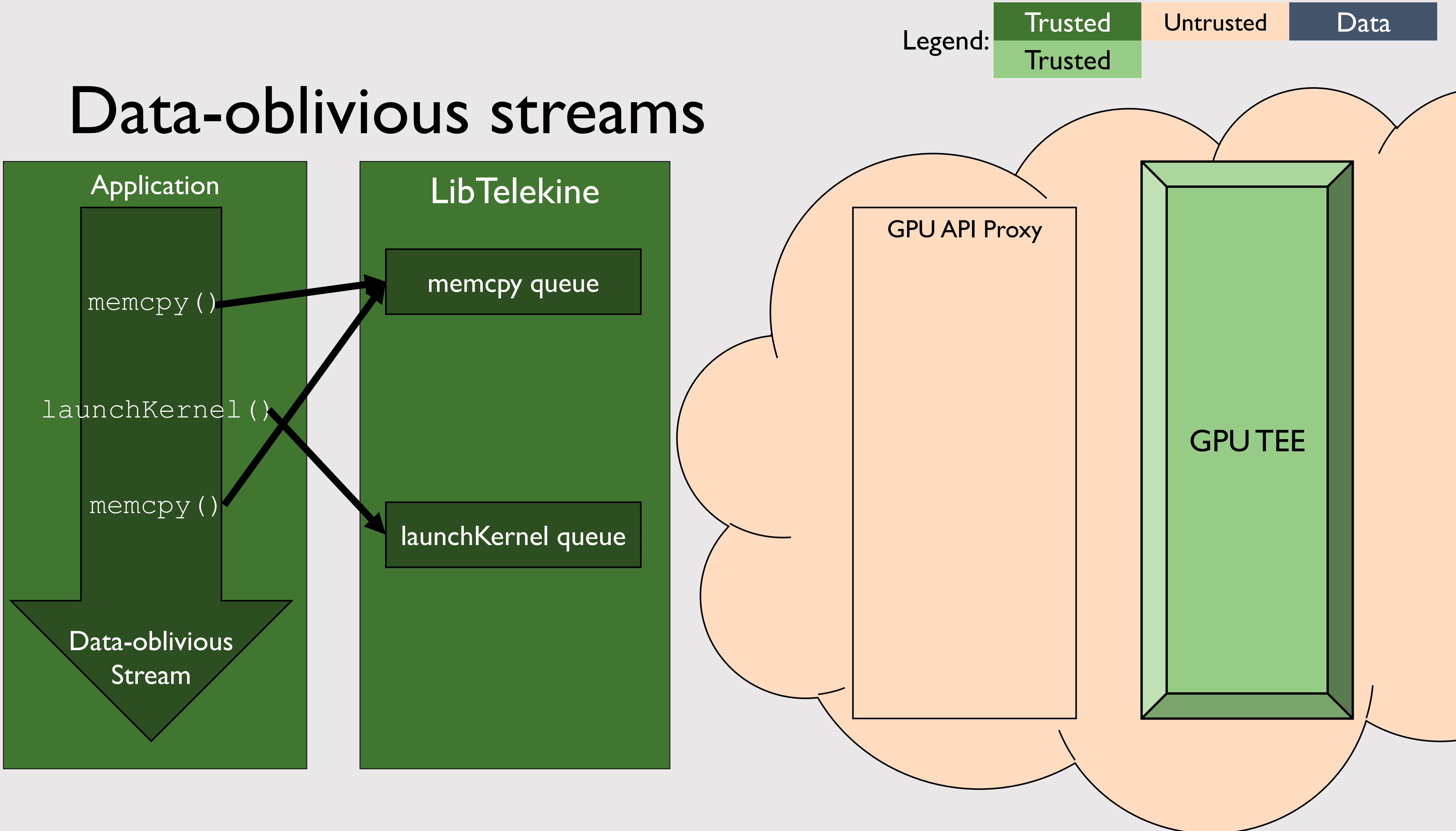
Legend: Trusted | Untrusted | Data | Trusted



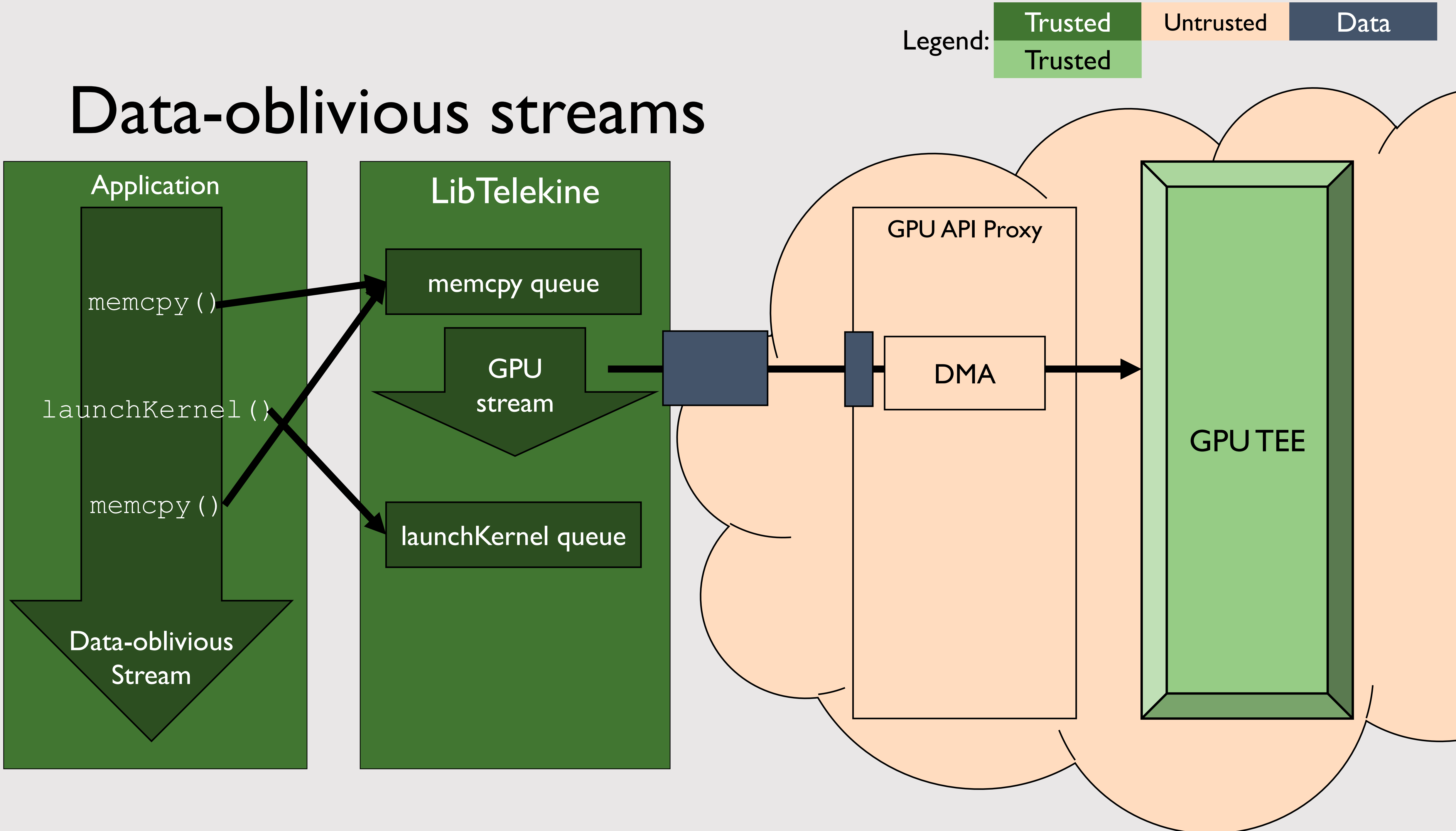
Data-oblivious streams



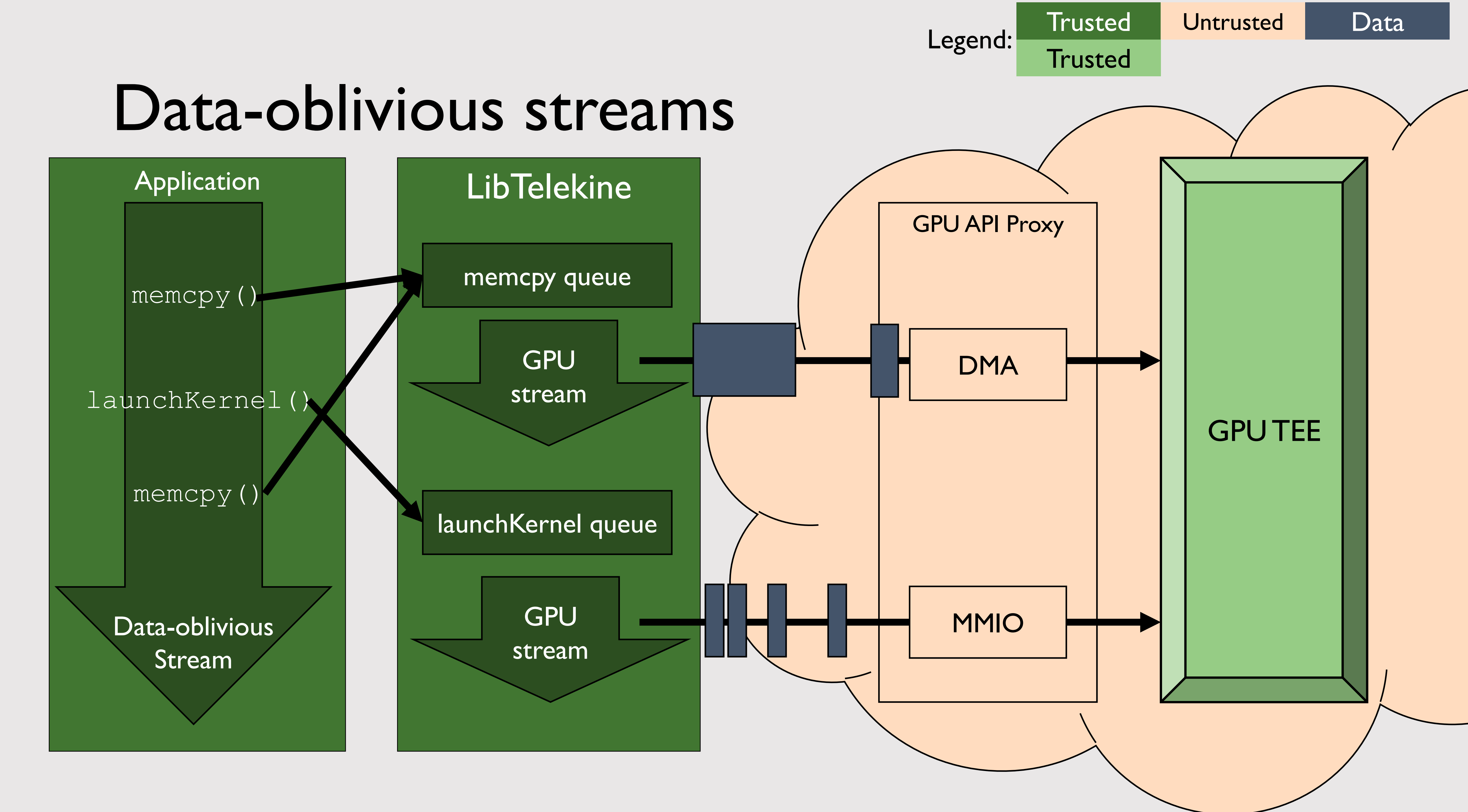
Data-oblivious streams



Data-oblivious streams



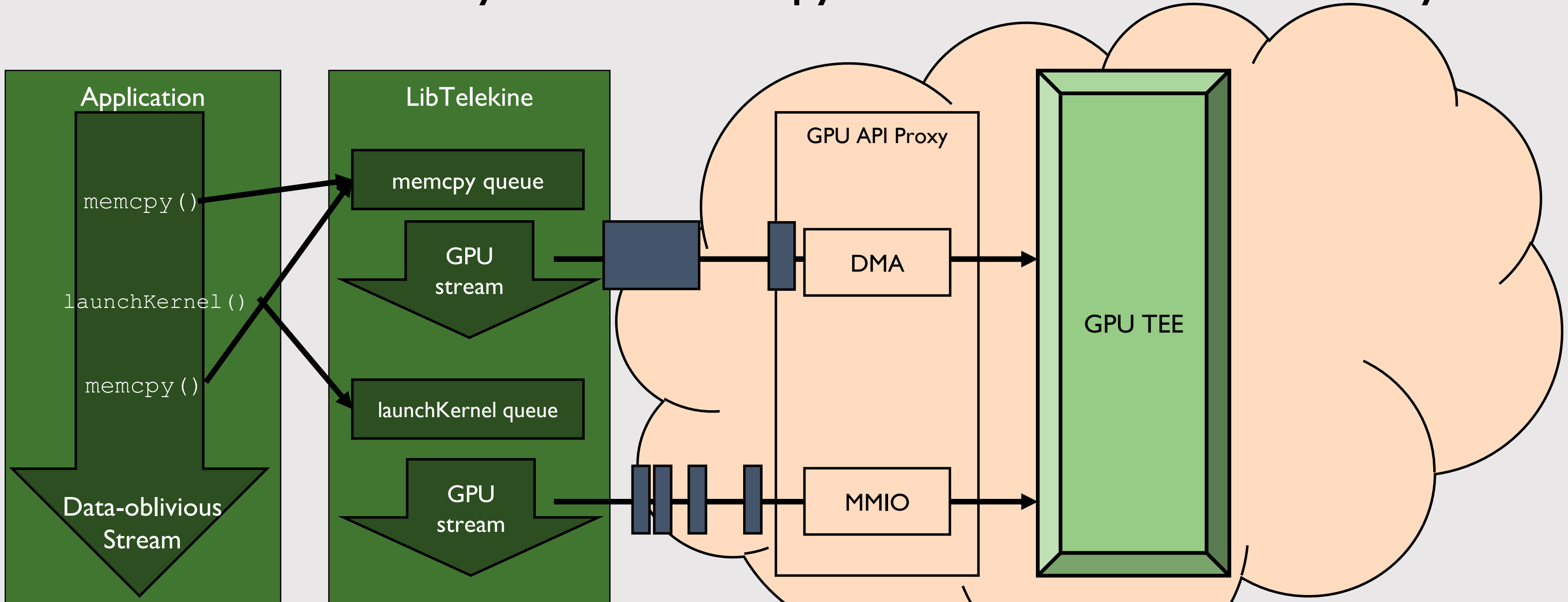
Data-oblivious streams



Legend: Trusted | Untrusted | Data
Trusted

Data-oblivious streams

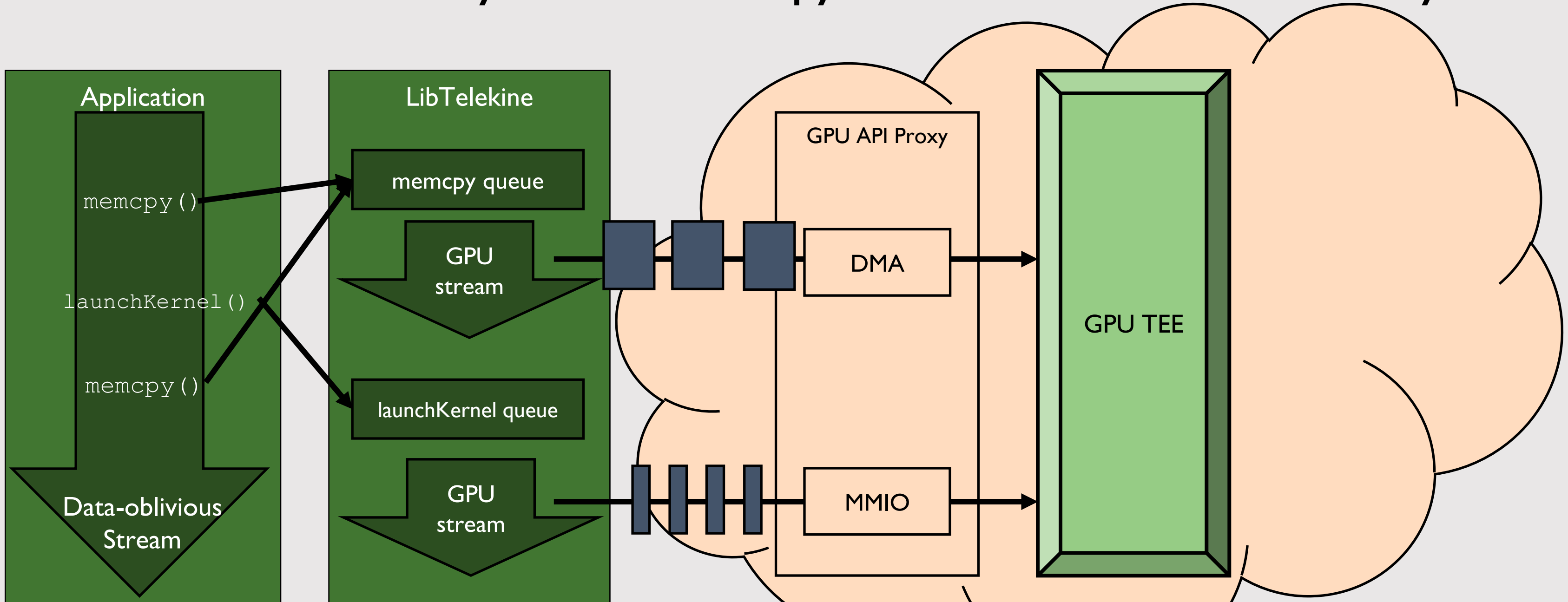
- Divide commands by type so they can be scheduled independently
  - Adversary sees two independent streams of operations
  - Telekine manages data dependencies between types
- Split and pad commands as necessary; enforce a uniform size
- Queue commands and send them (or no-ops) out deterministically
  - E.g., launch 32 kernels every 15ms, memcpy 1MB both directions every 30ms



Legend: Trusted | Trusted | Untrusted | Data

Data-oblivious streams

- Divide commands by type so they can be scheduled independently
  - Adversary sees two independent streams of operations
  - Telekine manages data dependencies between types
- Split and pad commands as necessary; enforce a uniform size
- Queue commands and send them (or no-ops) out deterministically
  - E.g., launch 32 kernels every 15ms, memcpy 1MB both directions every 30ms



In the rest of the talk we will answer:

- Can information be extracted from GPU communication patterns?

  Yes, we demonstrate a communication timing attack
- How does Telekine remove that information?

  Replace GPU streams with new data-oblivious streams
- What are Telekine's overheads?

  Overheads are reasonable: ~20% for neural network training

In the rest of the talk we will answer:

- Can information be extracted from GPU communication patterns?

  Yes, we demonstrate a communication timing attack

- How does Telekine remove that information?

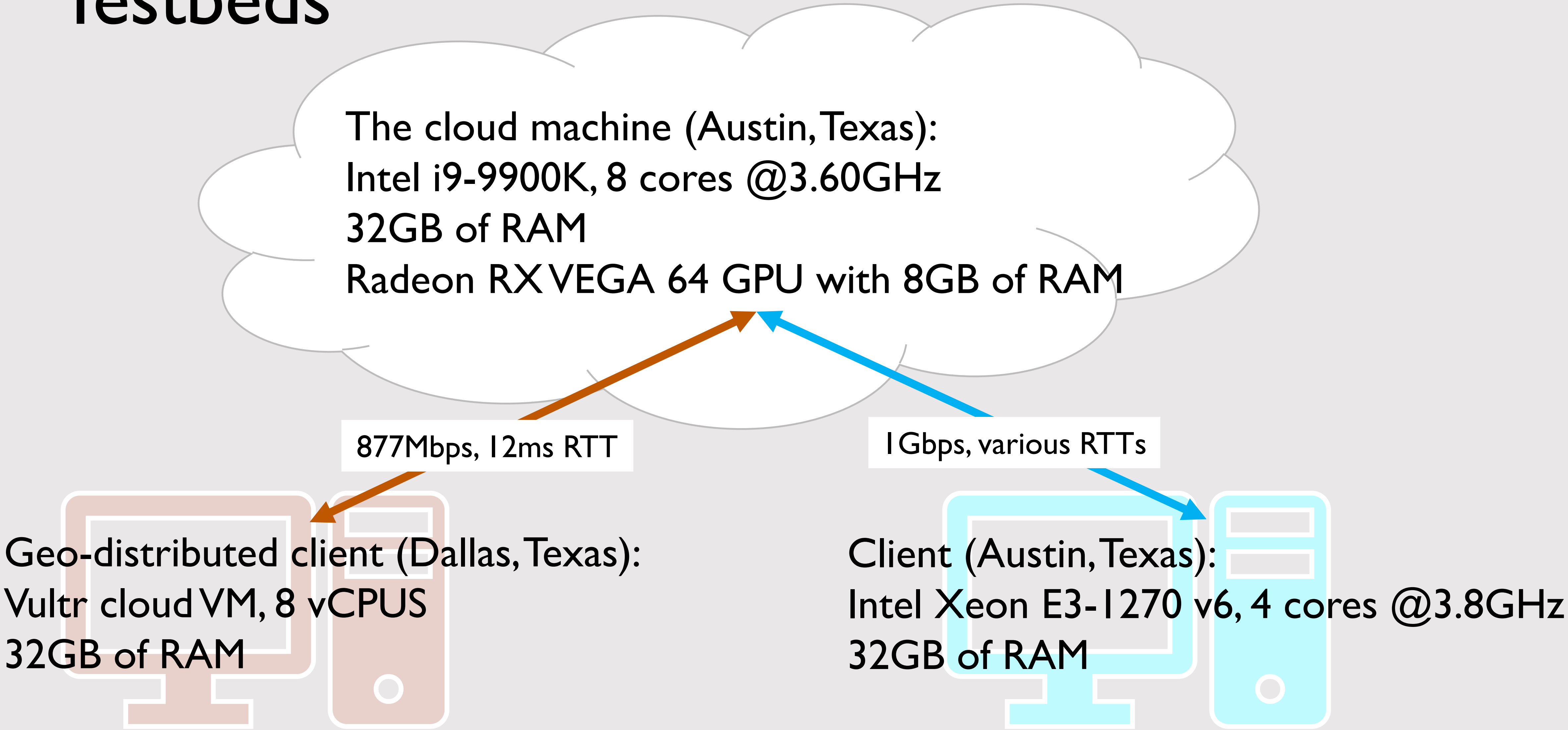
  Replace GPU streams with new data-oblivious streams

- What are Telekine's overheads?

  Overheads are reasonable: ~20% for neural network training

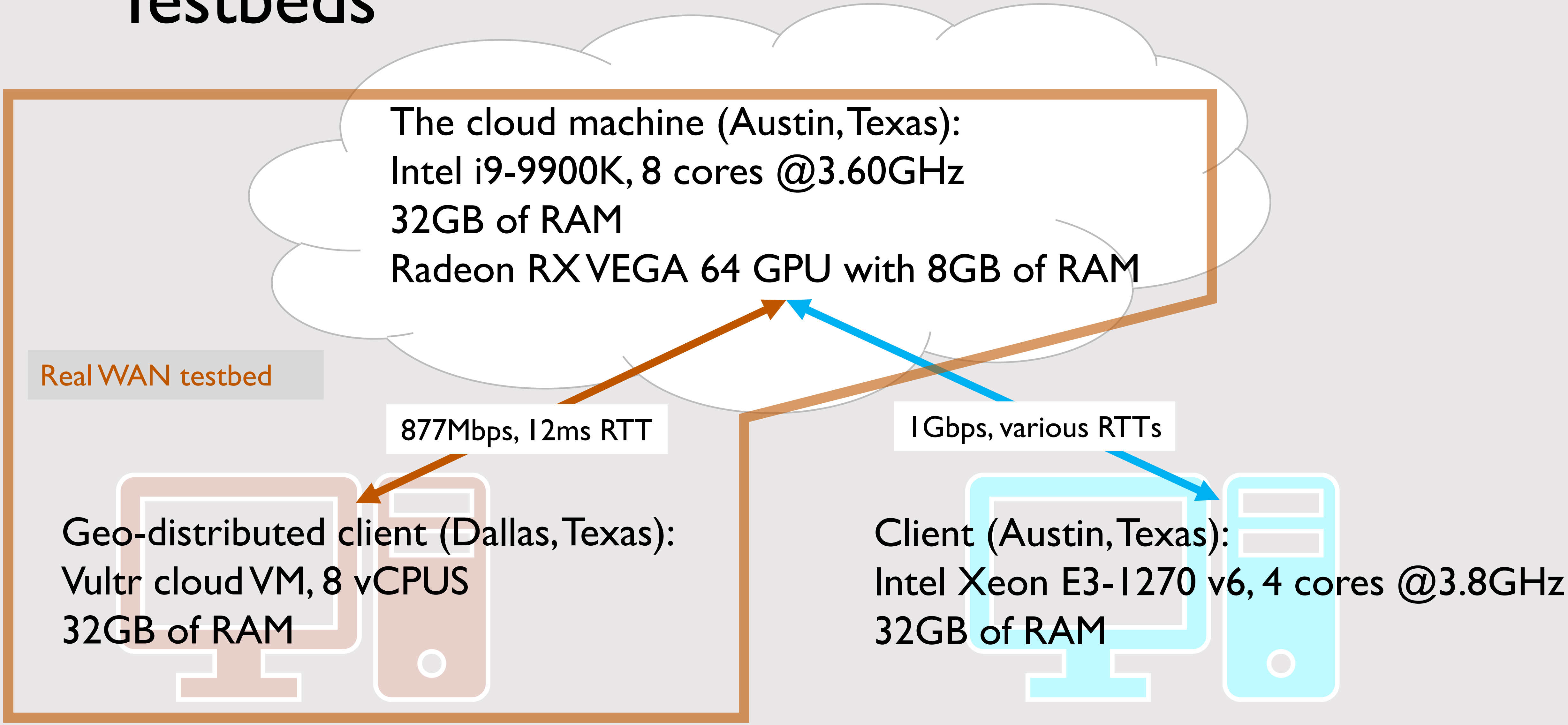
Testbeds

(RTT is “Roundtrip Time”)



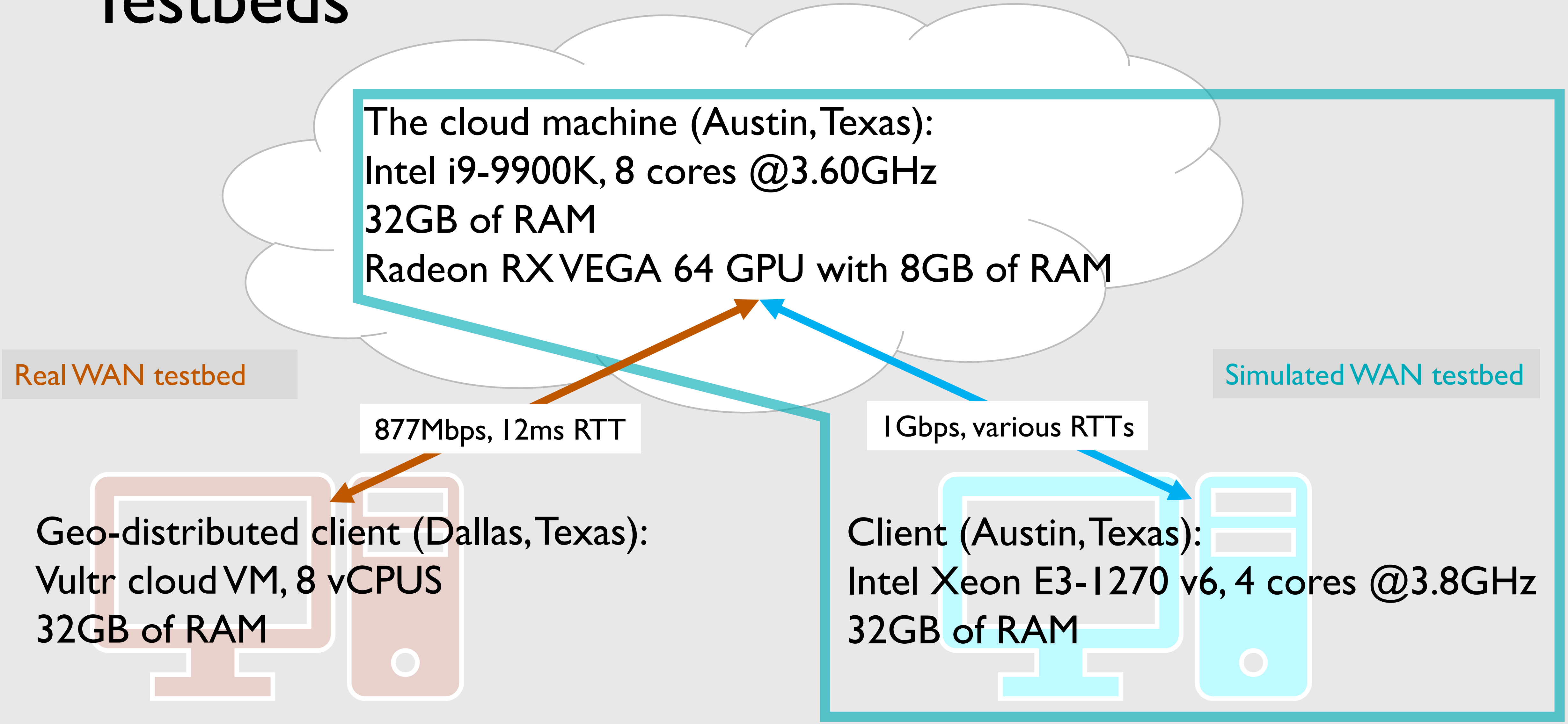
Testbeds

(RTT is “Roundtrip Time”)



Testbeds

(RTT is “Roundtrip Time”)



We compare Telekine to an insecure baseline: running on the GPU server without protections

Workloads:

- Data movement vs. GPU work microbenchmark
- Neural net inference on MXNet:
  - ResNet50[He et. al 2016], InceptionV3[Szegedy et. al 2016], DenseNet [Huang et. al 2017]
- Neural net training on MXNet:
  - (Same networks as above)
- Graph analytics on Galois:
  - BFS, PageRank, SSSP (across 1 and 2 GPUs)

We compare Telekine to an insecure baseline: running on the GPU server without protections

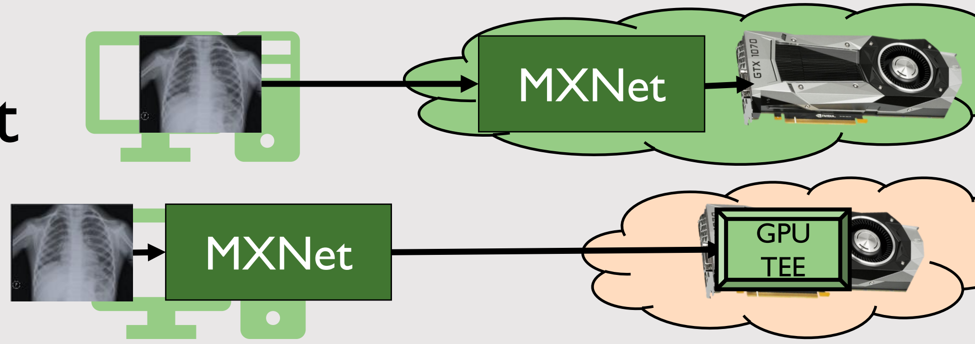
Workloads:

- Data movement vs. GPU work microbenchmark
- Neural net inference on MXNet:
  - ResNet50[He et. al 2016], InceptionV3[Szegedy et. al 2016], DenseNet [Huang et. al 2017]
- Neural net training on MXNet:
  - (Same networks as above)
- Graph analytics on Galois:
  - BFS, PageRank, SSSP (across 1 and 2 GPUs)

Legend: Trusted | Untrusted | Data
Trusted

MXNet neural net inference (Real WAN)

- User sends a batch of images to be classified
- Baseline: user sends batch to remote MXNet
- Telekine: user sends batch to local MXNet
  - Telekine remotes computation to the GPU

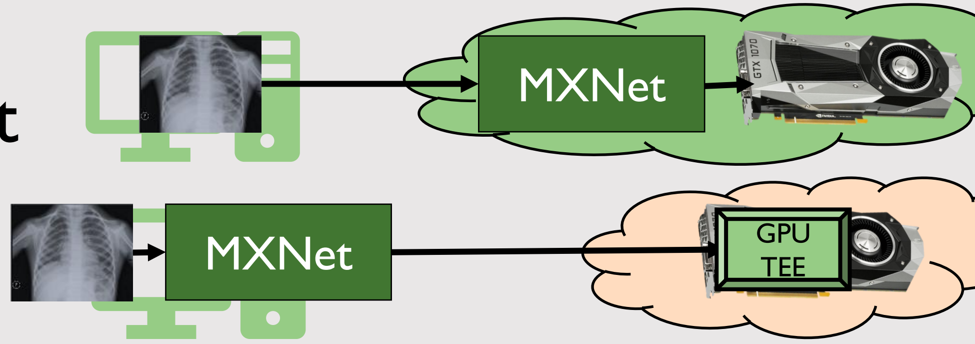


Batch Size	ResNet50	InceptionV3	DenseNet
1			
8			
64			

Legend: Trusted | Untrusted | Data | Trusted

MXNet neural net inference (Real WAN)

- User sends a batch of images to be classified
- Baseline: user sends batch to remote MXNet
- Telekine: user sends batch to local MXNet
    - Telekine remotes computation to the GPU

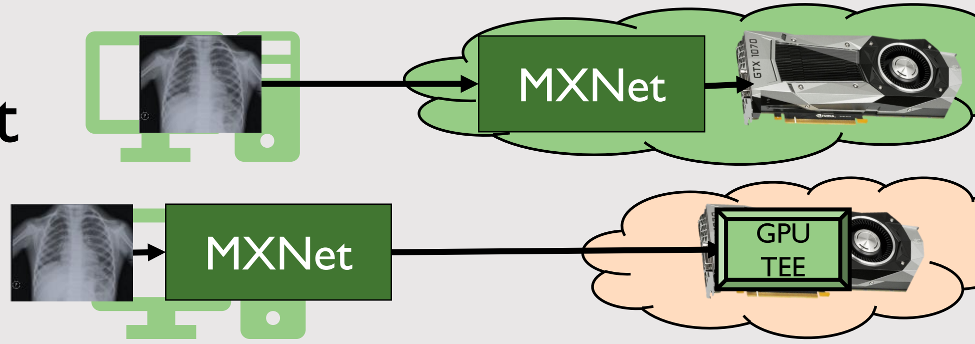


Batch Size	ResNet50	InceptionV3	DenseNet
1	10.0X	6.6X	7.7X
8			
64			

Legend: Trusted | Untrusted | Data | Trusted

MXNet neural net inference (Real WAN)

- User sends a batch of images to be classified
- Baseline: user sends batch to remote MXNet
- Telekine: user sends batch to local MXNet
  - Telekine remotes computation to the GPU

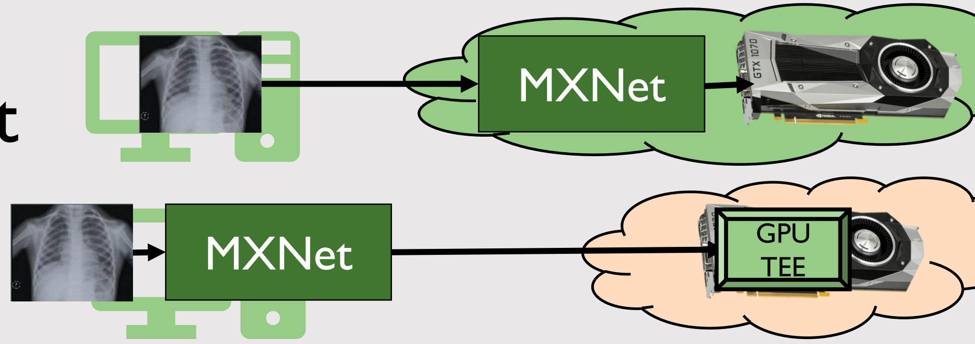


Batch Size	ResNet50	InceptionV3	DenseNet
1	10.0X	6.6X	7.7X
8	3.4X	2.2X	2.5X
64			

Legend: Trusted | Untrusted | Data
Trusted

MXNet neural net inference (Real WAN)

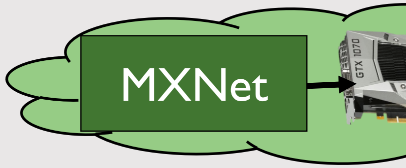
- User sends a batch of images to be classified
- Baseline: user sends batch to remote MXNet
- Telekine: user sends batch to local MXNet
  - Telekine remotes computation to the GPU



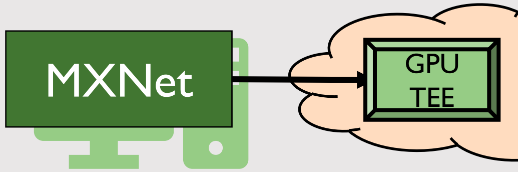
Batch Size	ResNet50	InceptionV3	DenseNet
1	10.0X	6.6X	7.7X
8	3.4X	2.2X	2.5X
64	1.0X	1.1X	1.0X

MXNet neural net training (Real WAN)

- Large dataset of images, processed in batches of size 64
- Baseline: data set is on **the cloud machine**, passed to MXNet



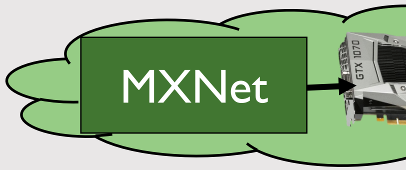
- Telekine: data set is on **a client**, passed to MXNet instance
  - Telekine connects that instance to the remote GPU
  - As a result Telekine uses a consistent 533 Mb/s network bandwidth



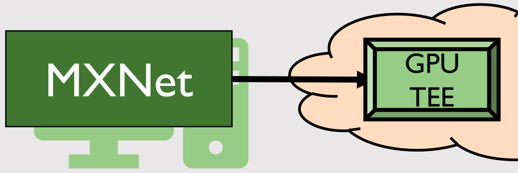
ResNet50	InceptionV3	DenseNet

MXNet neural net training (Real WAN)

- Large dataset of images, processed in batches of size 64
- Baseline: data set is on **the cloud machine**, passed to MXNet



- Telekine: data set is on **a client**, passed to MXNet instance
  - Telekine connects that instance to the remote GPU
  - As a result Telekine uses a consistent 533 Mb/s network bandwidth



ResNet50	InceptionV3	DenseNet
1.23X	1.08X	1.22X

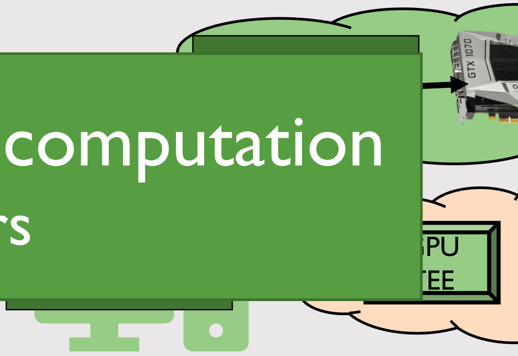
MXNet neural net training (Real WAN)

- Large dataset of images, processed in batches of size 64

Overheads are low because GPUs overlap the extra work with computation

- E.g., CUs can keep processing while the DMA engine performs transfers

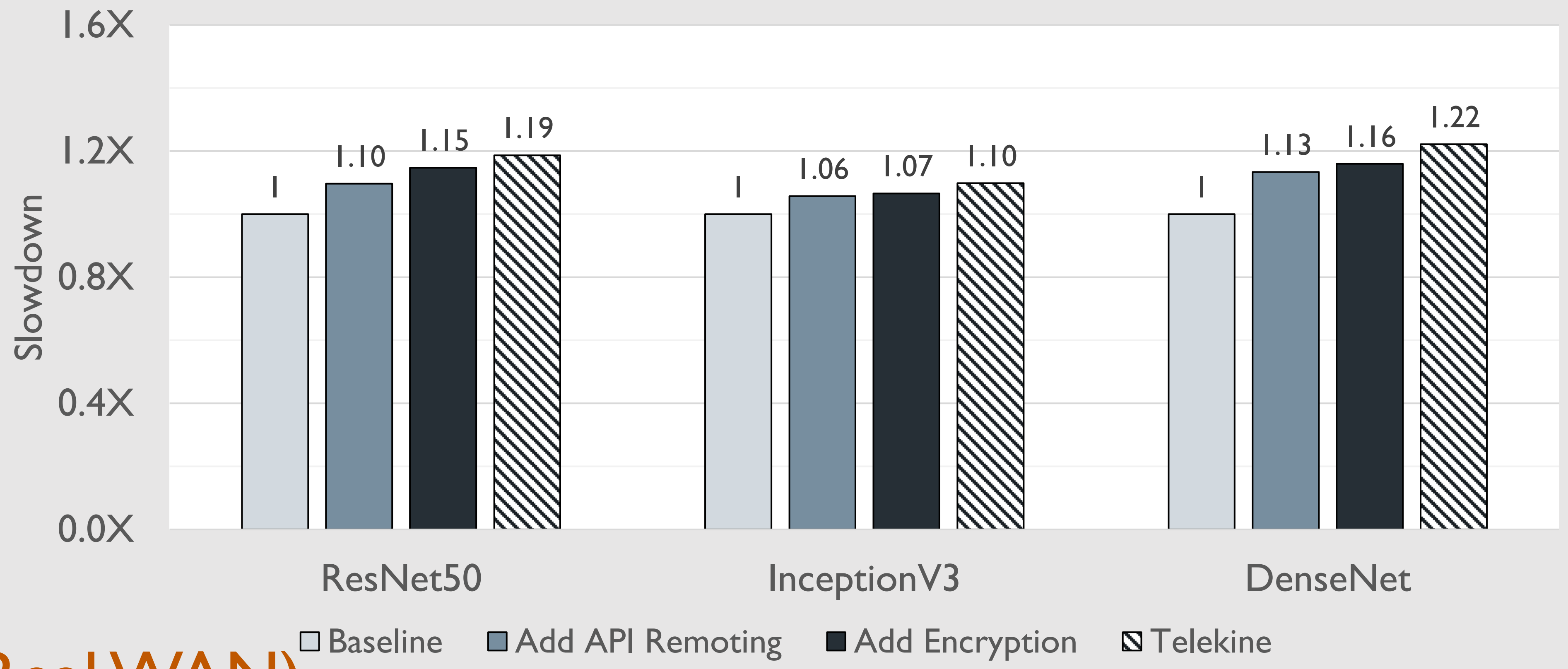
  - Telekine connects that instance to the remote GPU
  - As a result Telekine uses a consistent 533 Mb/s network bandwidth



ResNet50	InceptionV3	DenseNet
1.23X	1.08X	1.22X

MXNet neural net training breakdown (Simulated WAN)

10ms RTT

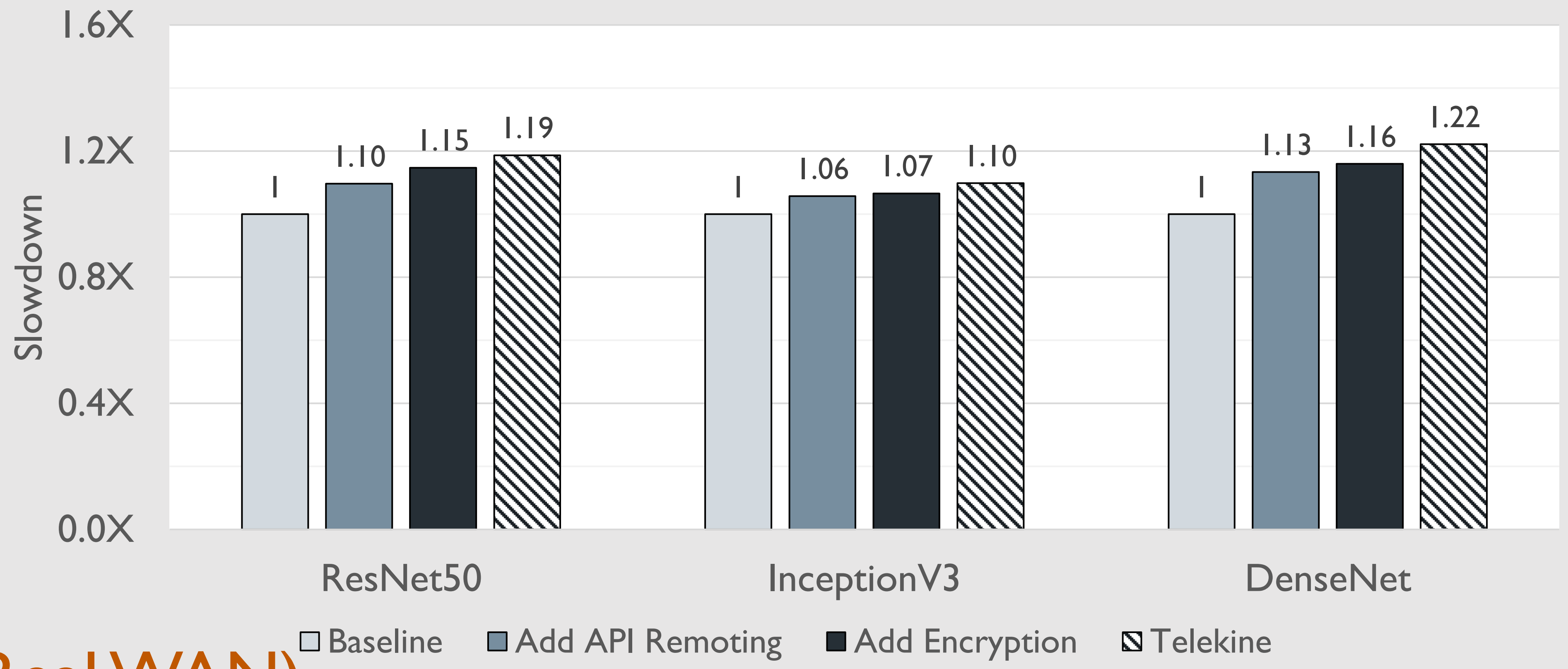


(Real WAN)

ResNet50	InceptionV3	DenseNet
1.23X	1.08X	1.22X

MXNet neural net training breakdown (Simulated WAN)

10ms RTT

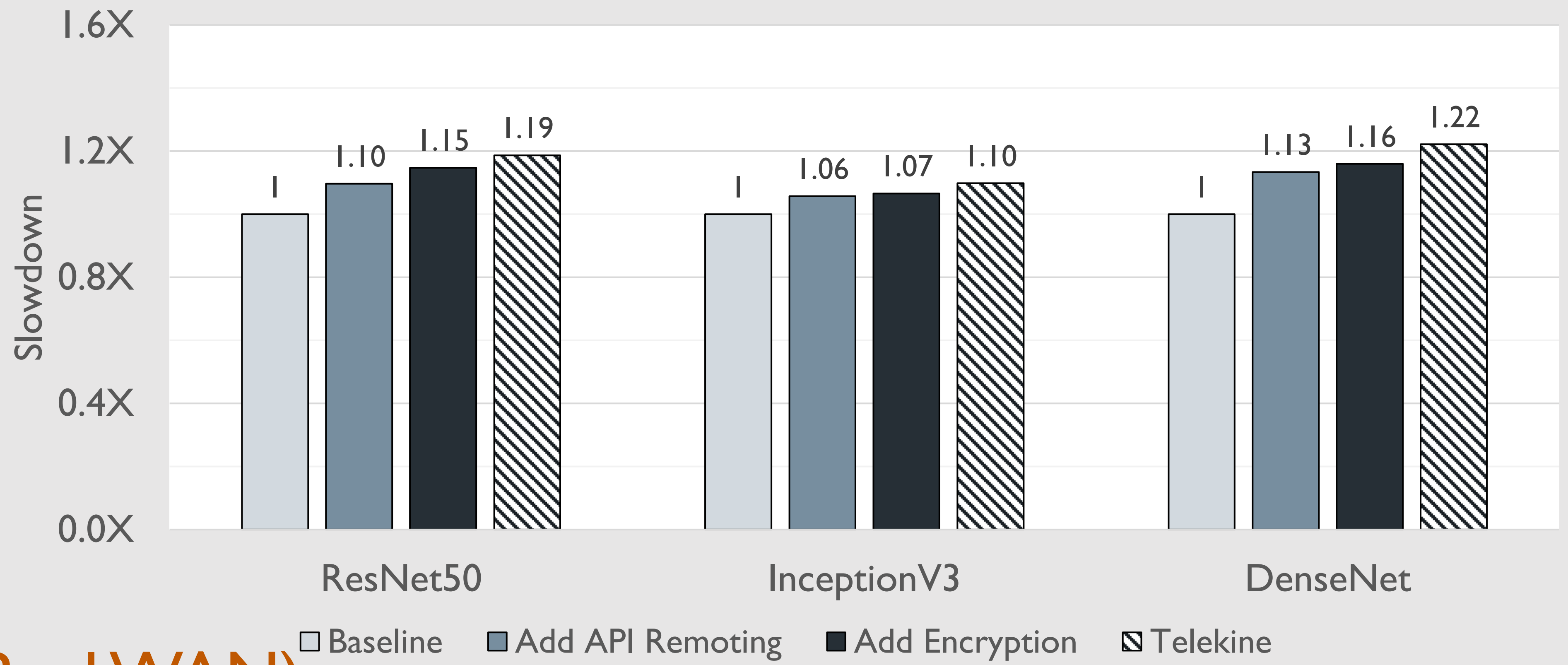


(Real WAN)

ResNet50	InceptionV3	DenseNet
1.23X	1.08X	1.22X

MXNet neural net training breakdown (Simulated WAN)

10ms RTT

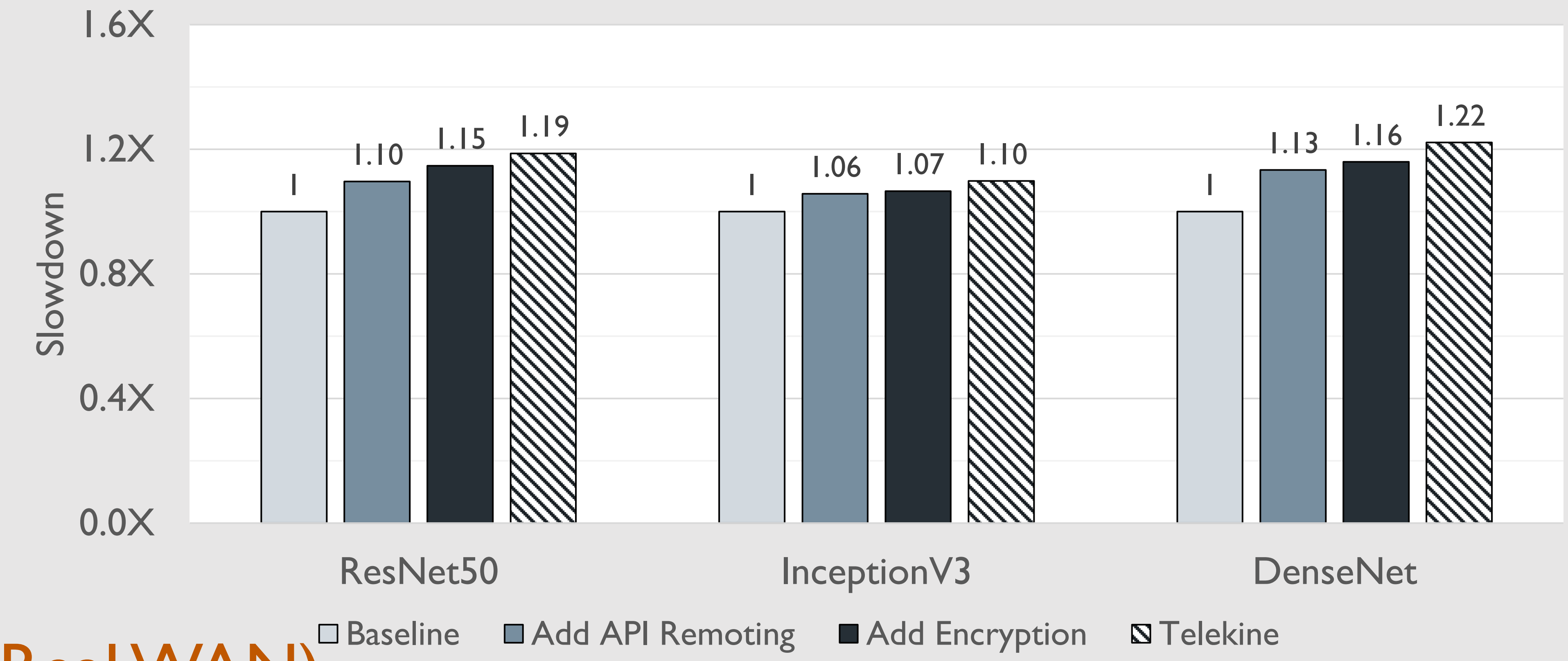


(Real WAN)

ResNet50	InceptionV3	DenseNet
1.23X	1.08X	1.22X

MXNet neural net training breakdown (Simulated WAN)

10ms RTT



(Real WAN)

ResNet50	InceptionV3	DenseNet
1.23X	1.08X	1.22X

Telekine: Secure Computing with Cloud GPUs

- Eliminates communication timing channels with data-oblivious streams
- Transparent to applications because it maintains GPU API semantics
- Has modest performance overheads for level of security provided

Telekine: Secure Computing with Cloud GPUs

- Eliminates communication timing channels with data-oblivious streams
- Transparent to applications because it maintains GPU API semantics
- Has modest performance overheads for level of security provided

Thanks!

Backup slides follow

MXNet training RTT sensitivity (Simulated WAN)

- RTT to cloud provider can vary
- The effect in performance depends on the workload

RTT	ResNet50	InceptionV3	Densenet
10ms	1.19X	1.10X	1.22X
20ms	1.29X	1.13X	1.37X
30ms	1.44X	1.16X	1.49X
40ms	1.53X	1.18X	1.66X
50ms	1.62X	1.30X	2.09X

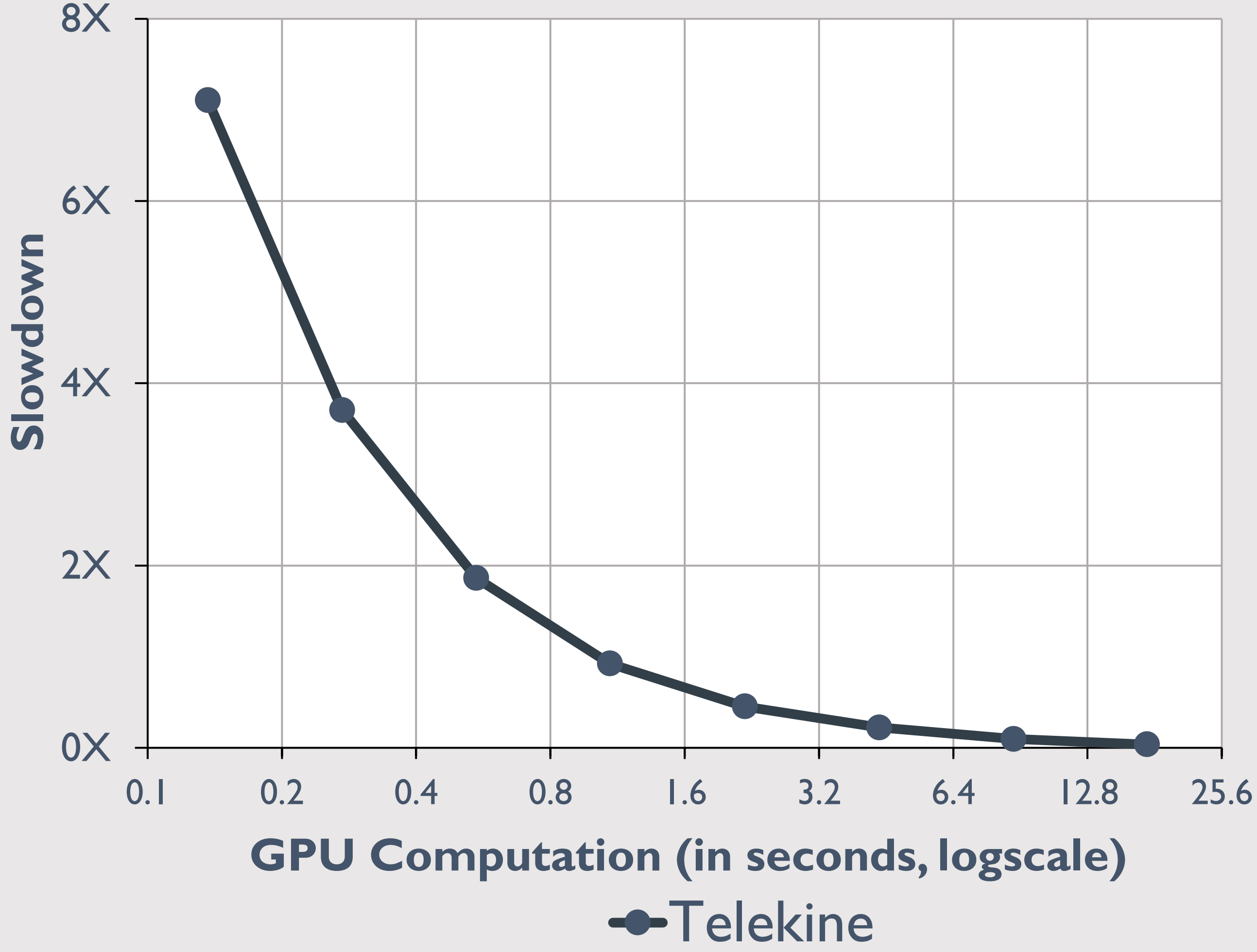
Attack accuracy for batched inference

- GPU kernels operate on an entire batch
  - Cannot measure kernel execution time for individual images
- Task: correctly identify the class with the most images
  - Accuracy varies with how many more images there are (purity)
  - Batches of 32, four classes
  - Images selected from target class up to "Purity"
  - Batch filled out with images from other three classes

Batch size	Purity	Accuracy
1	100%	42%
32	25%	29%
32	80%	50%
32	100%	65%

Communication vs GPU work (Simulated WAN)

10ms RTT



- Copy16MB to the GPU
- Compute for x-axis seconds
- Copy16MB from the GPU