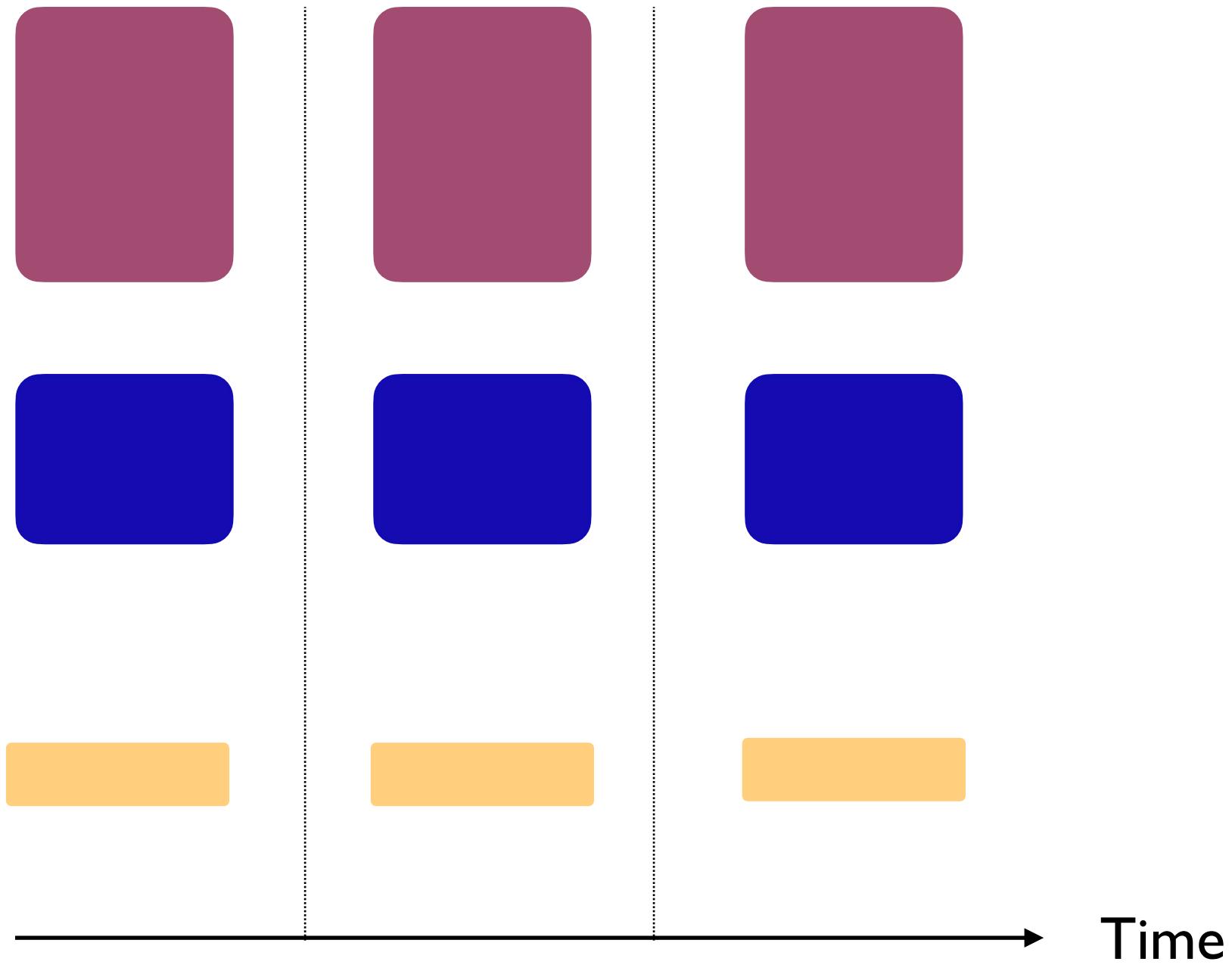


Future Internet 2021

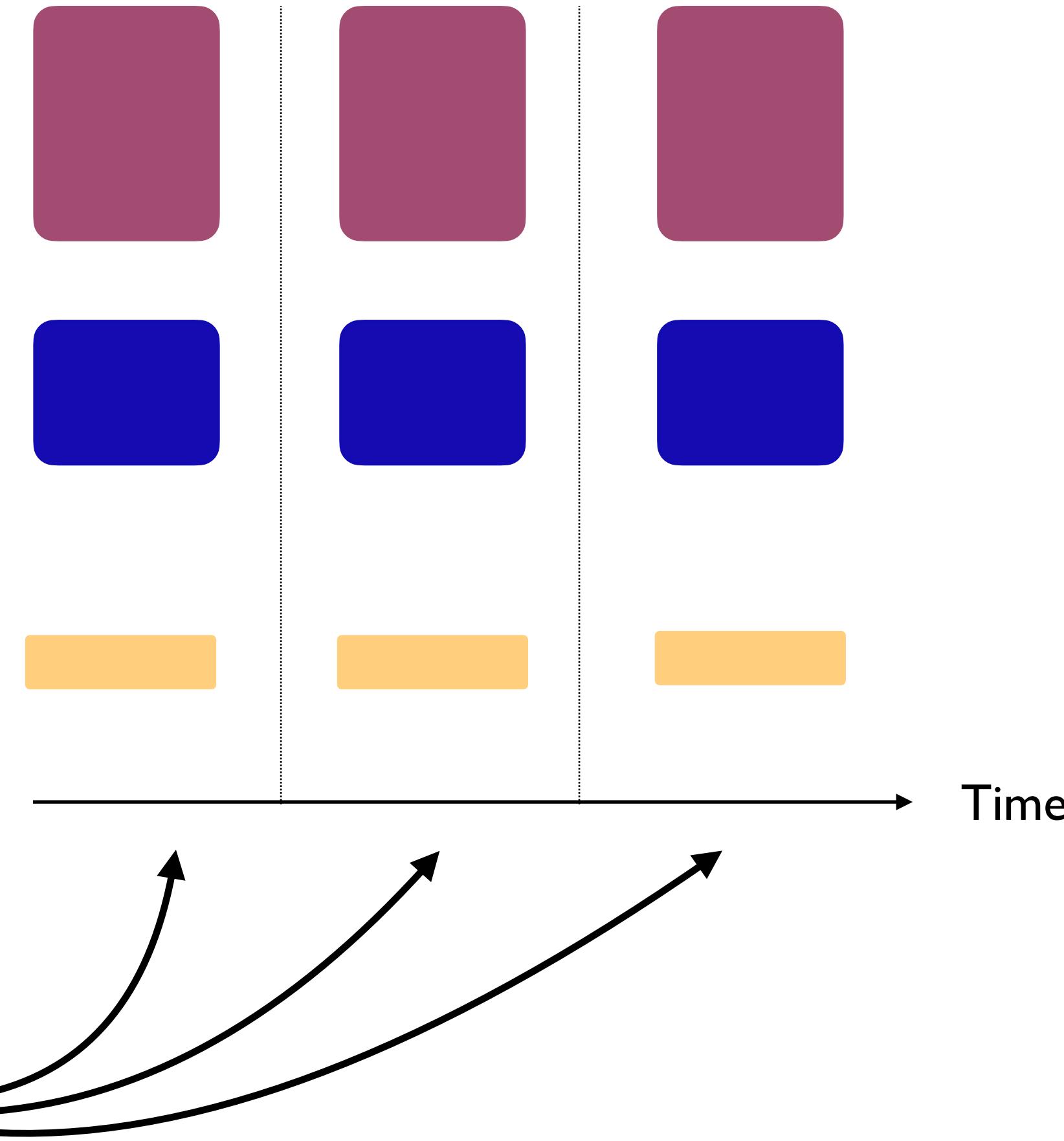
Project 2: Video Streaming

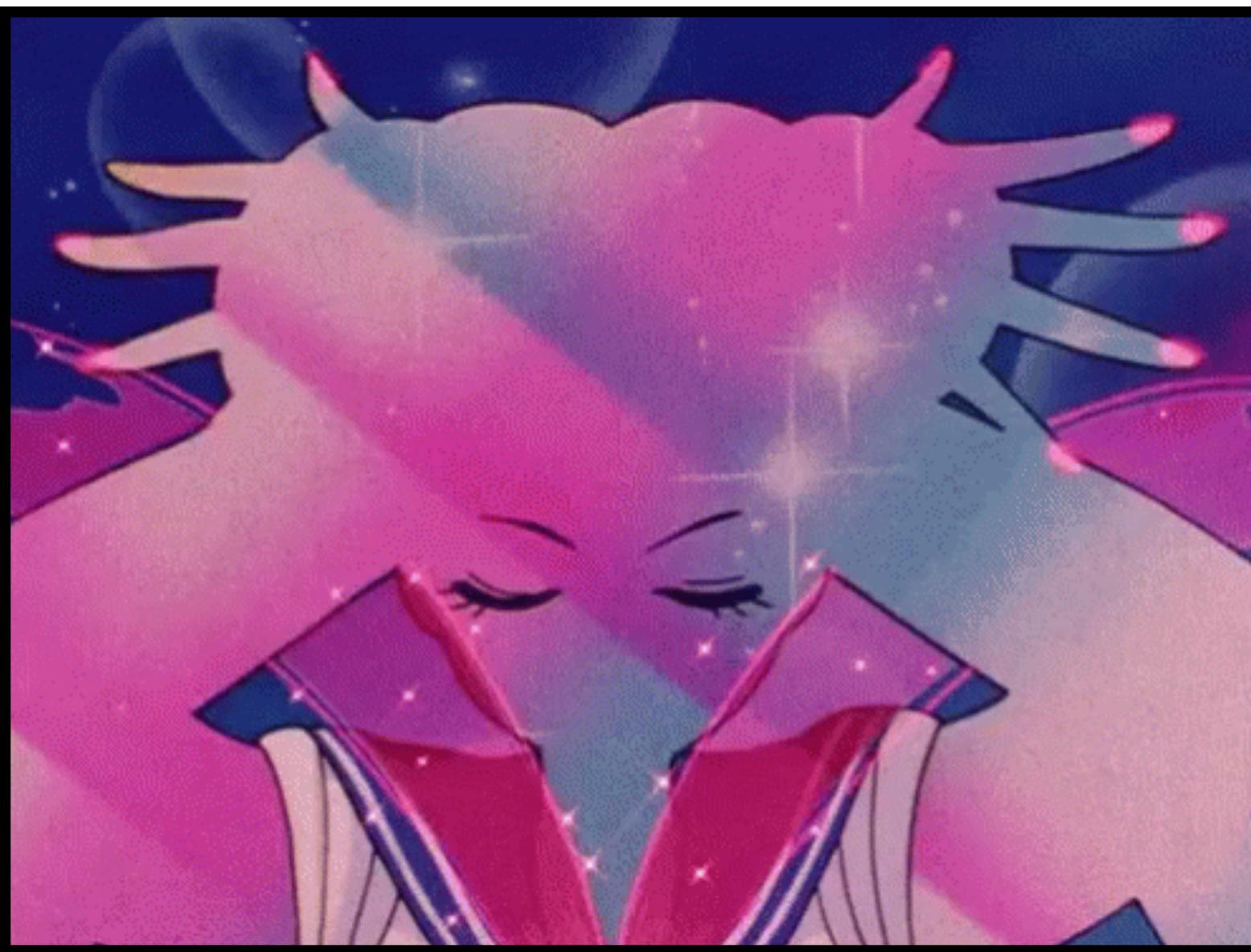
Melissa Licciardello, 30.03.2021



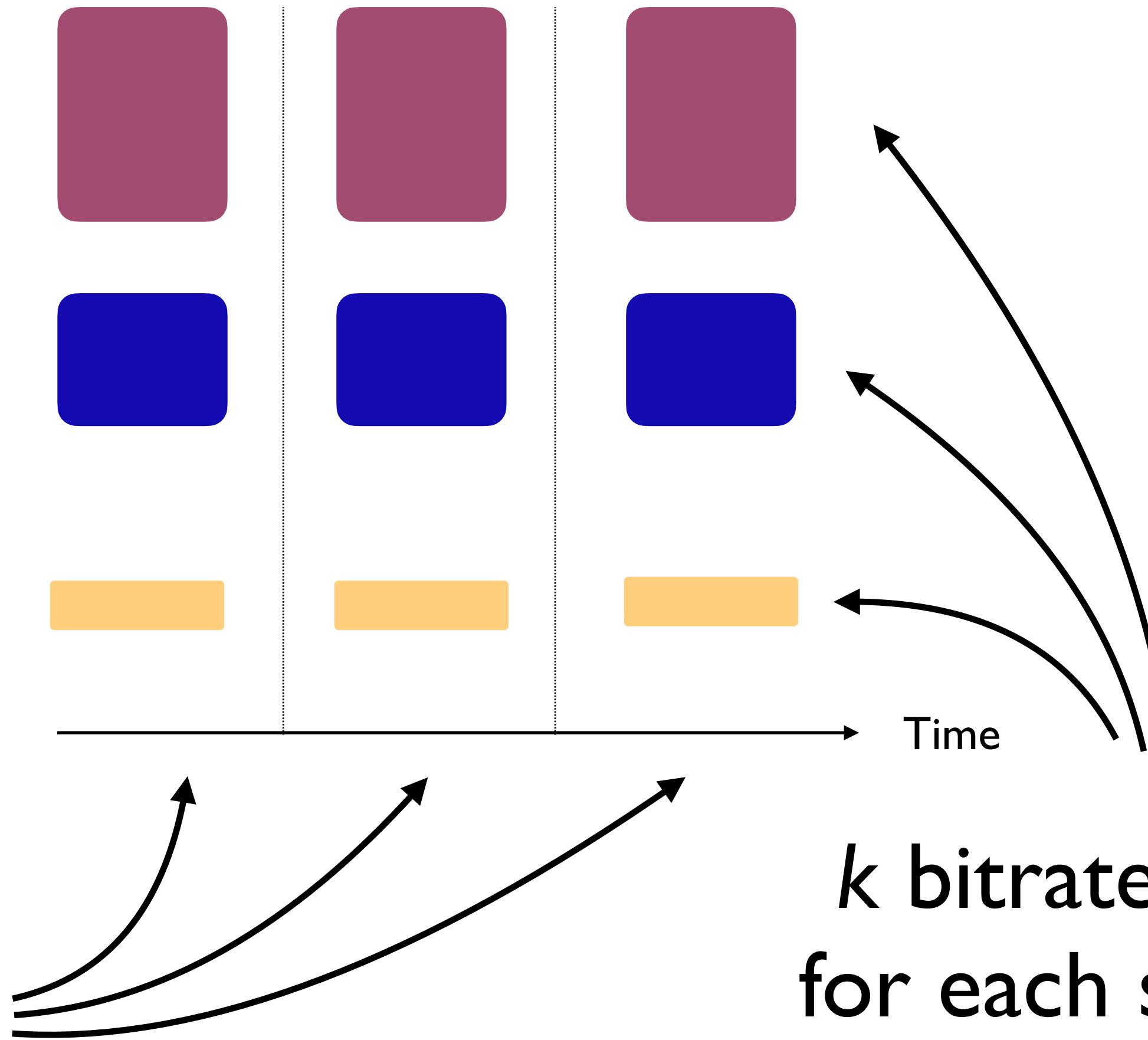


n second segments

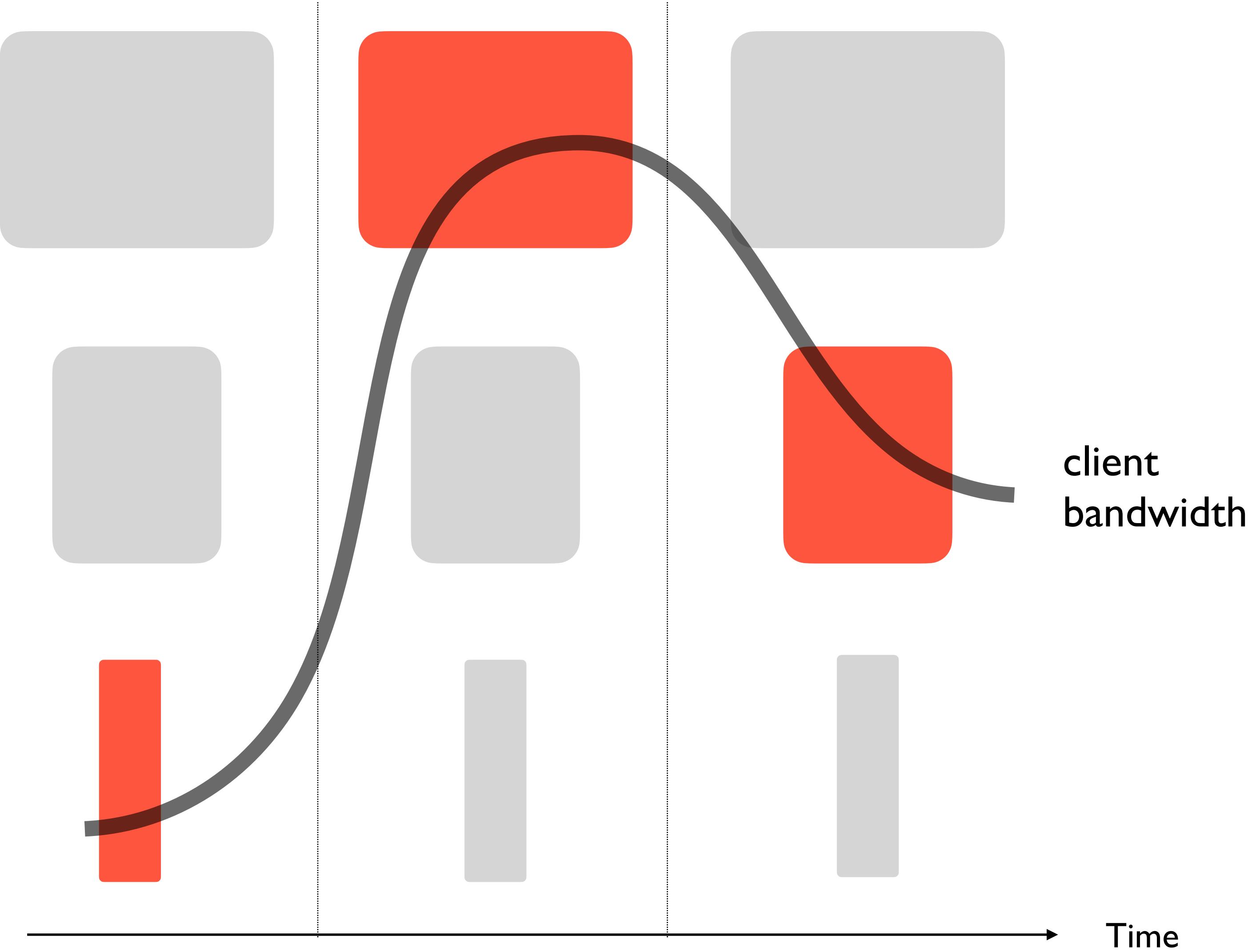




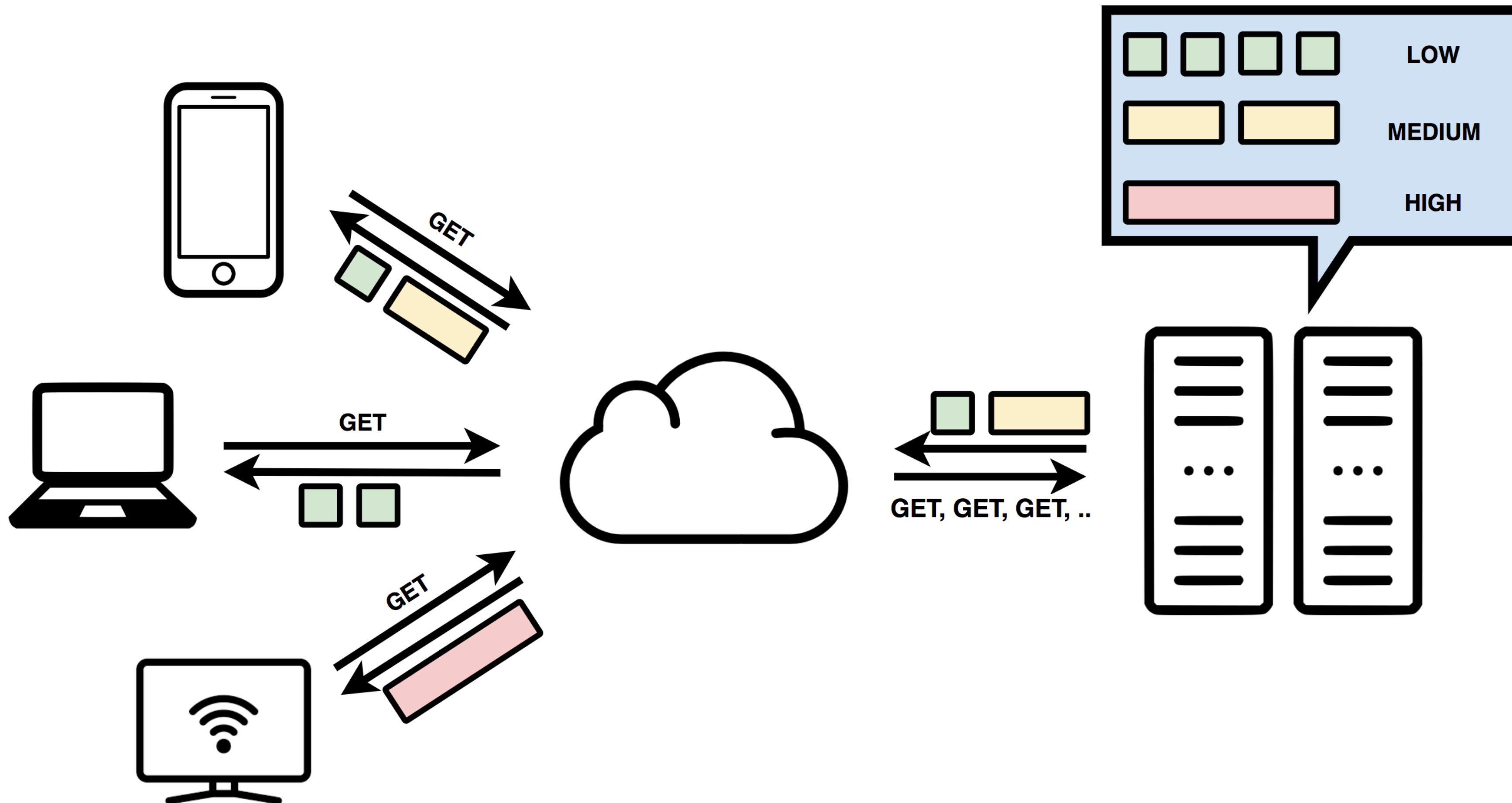
n second segments



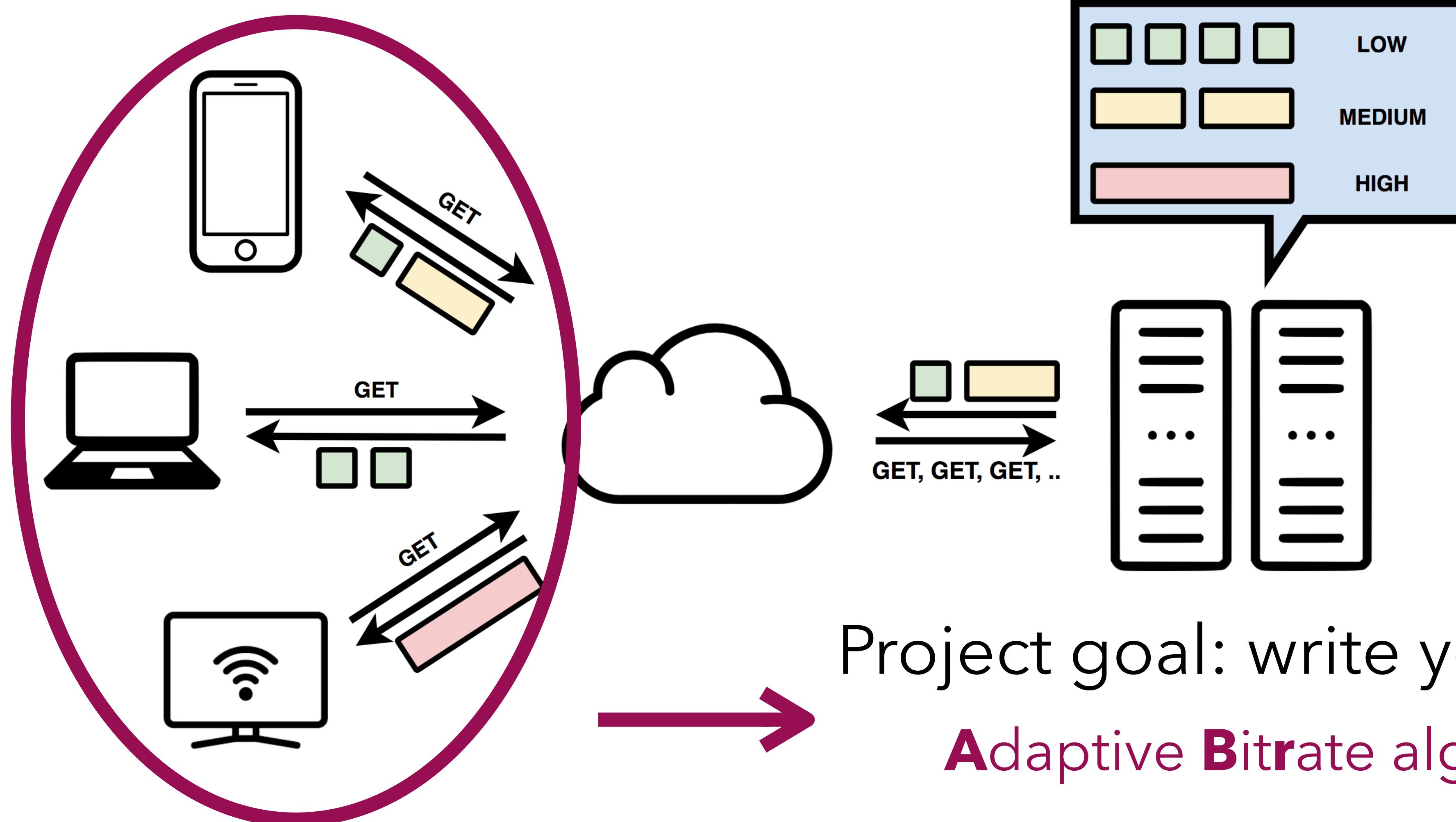
k bitrate tracks
for each segment



DASH architecture



DASH architecture

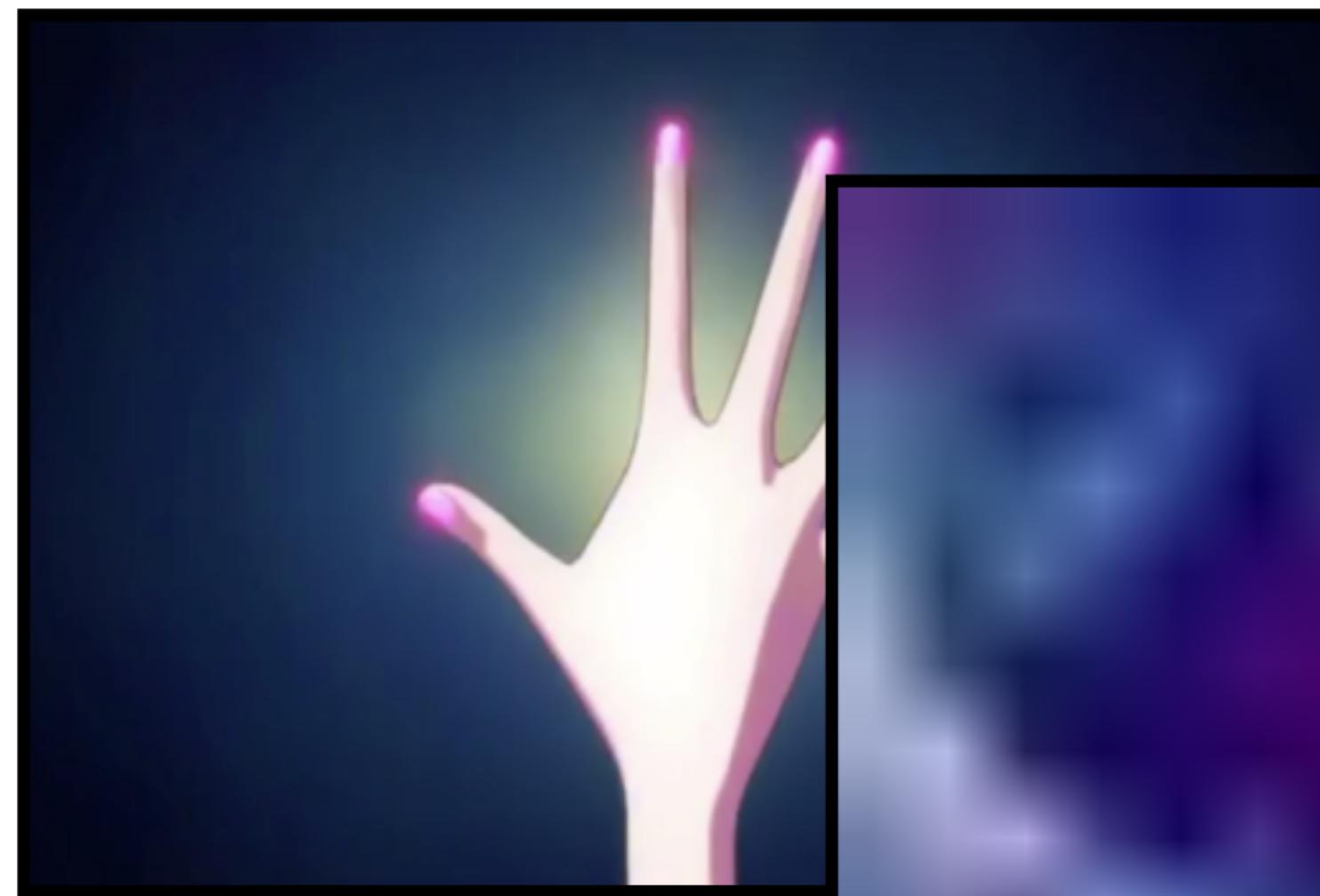


Project goal: write your **ABR**
Adaptive Bitrate algorithm





$t_0 : 720p$



$t_0 + dt_1 : 144p$



$t_0 + dt_2 : 360p$



$t_0 + dt_3 : 240p$



$t_0 + dt_4 : 1080p$



PLAYBACK

QoE function

QoE function

- Available bitrates (B_r): **0.3, 0.75, 1.2, 1.85, 2.85, 4.3** [MB]

QoE function

- Available bitrates (Br): **0.3, 0.75, 1.2, 1.85, 2.85, 4.3** [MB]
- **N** video chunks

QoE function

- Available bitrates (Br): **0.3, 0.75, 1.2, 1.85, 2.85, 4.3** [MB]
- **N** video chunks

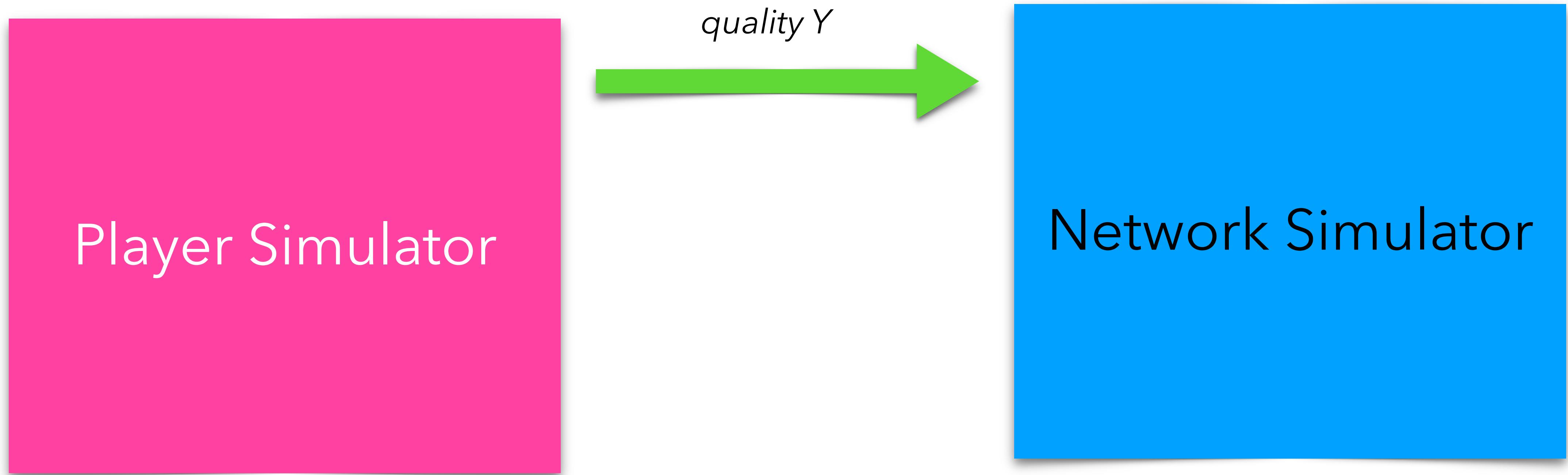
$$QoE = \sum_1^N Br_i - \sum_2^N |Br_i - Br_{i-1}| - 4.3 \cdot \sum_1^N \Delta_{rebuf}$$

Project abstraction

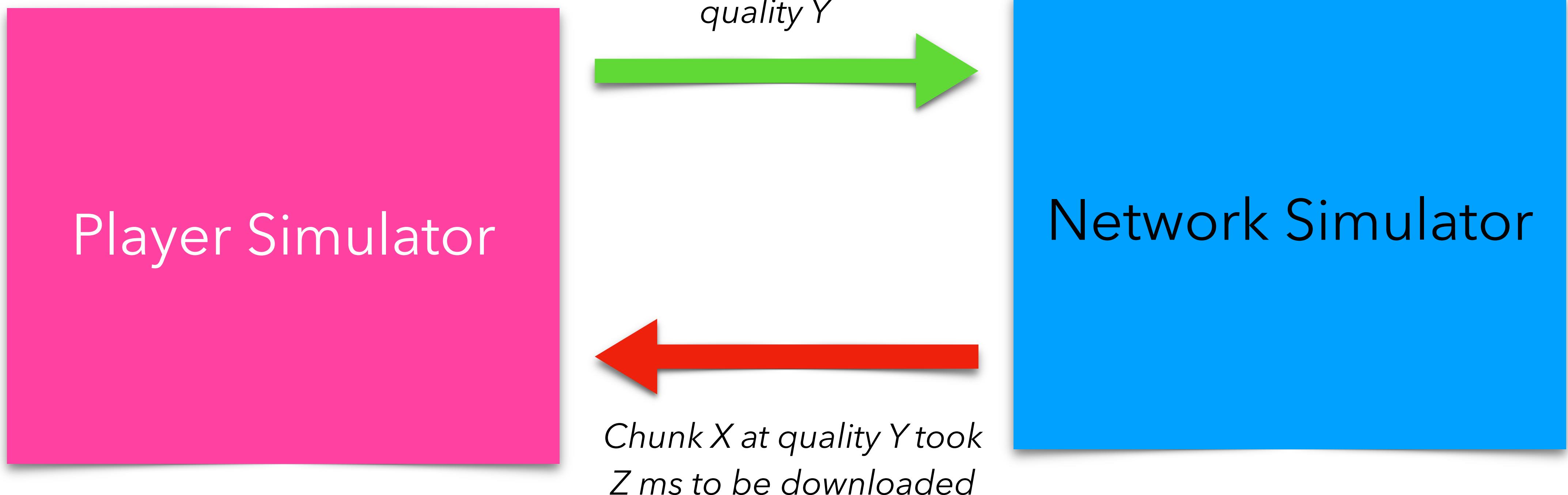
Player Simulator

Network Simulator

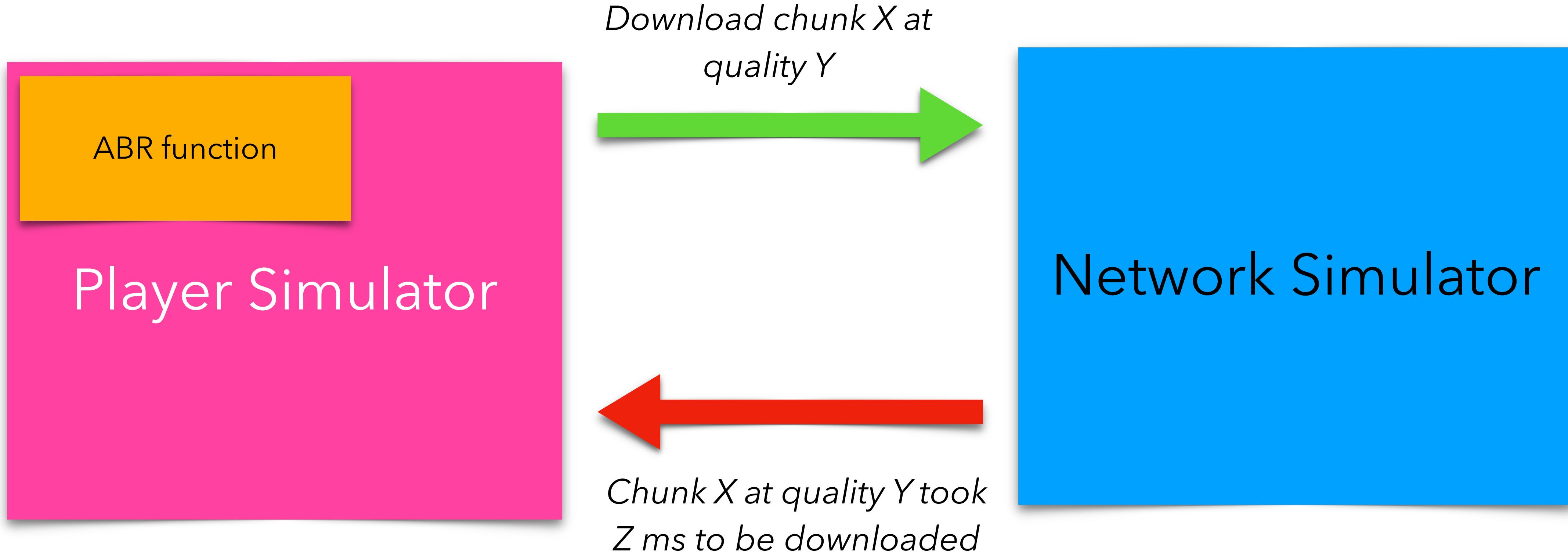
Project abstraction



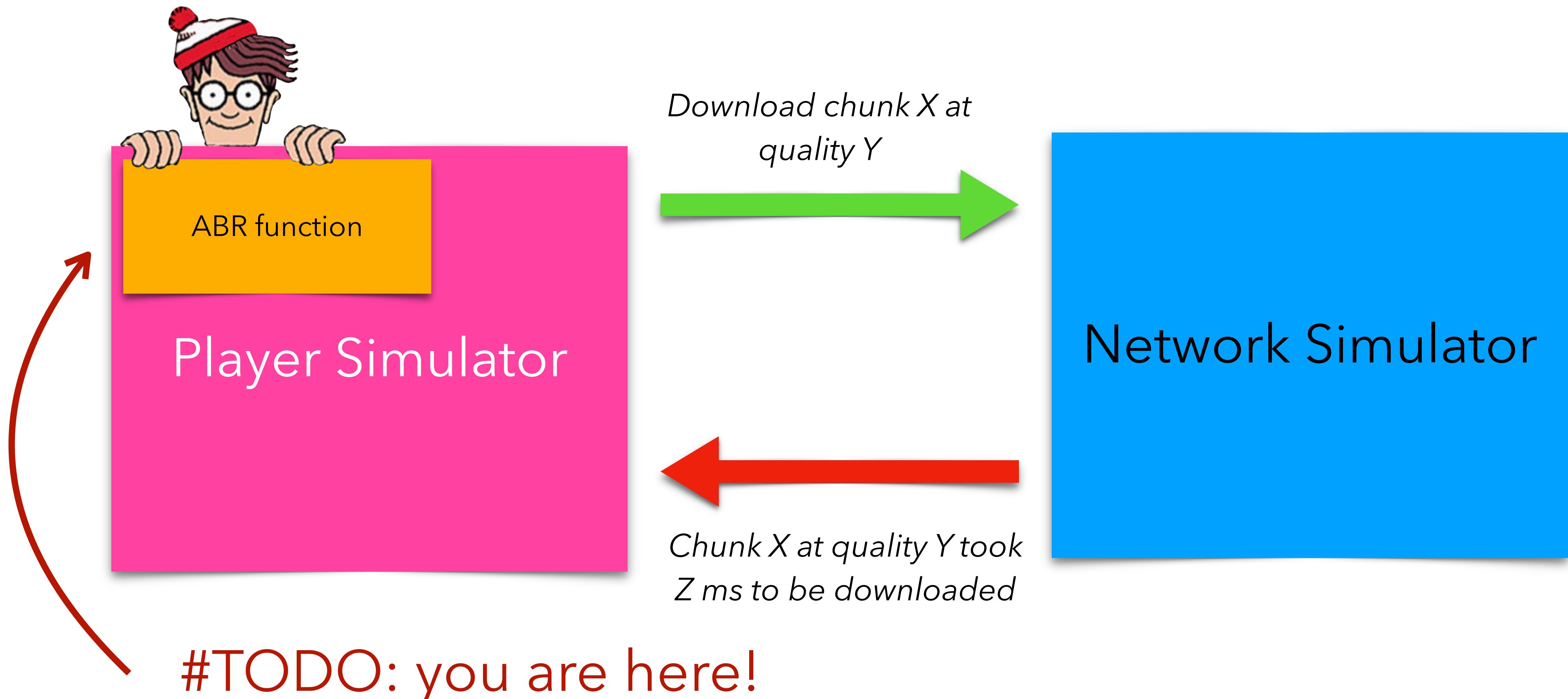
Project abstraction



Project abstraction



Project abstraction



When is my ABR function invoked in the code?

- **CALLBACK_EVENT.INIT**: Initial call at time 0

When is my ABR function invoked in the code?

- **CALLBACK_EVENT.INIT**: Initial call at time 0
- **CALLBACK_EVENT.DOWNLOAD_COMPLETED**: A chunk has been downloaded

When is my ABR function invoked in the code?

- **CALLBACK_EVENT.INIT**: Initial call at time 0
- **CALLBACK_EVENT.DOWNLOAD_COMPLETED**: A chunk has been downloaded
- **CALLBACK_EVENT.TIMEOUT**: Timeout has happened

When is my ABR function invoked in the code?

- **CALLBACK_EVENT.INIT**: Initial call at time 0
- **CALLBACK_EVENT.DOWNLOAD_COMPLETED**: A chunk has been downloaded
- **CALLBACK_EVENT.TIMEOUT**: Timeout has happened
- **CALLBACK_EVENT.REBUFFERING**: Rebuffering started

What does my ABR function outputs? (1)

1. Quality to download now → index from 0 to 5 ("space")



What does my ABR function outputs? (2)

2. Chunk index to download now → which video segment should be downloaded (“time”)

What does my ABR function outputs? (2)

2. Chunk index to download now → which video segment should be downloaded ("time")

 You can't download already played chunks.

What does my ABR function outputs? (2)

2. Chunk index to download now → which video segment should be downloaded ("time")

- You can't download already played chunks.
- If you return -1, no chunk will be downloaded

What does my ABR function outputs? (2)

2. Chunk index to download now → which video segment should be downloaded (“time”)

- You can't download already played chunks.
- If you return -1, no chunk will be downloaded
- If the previous download hasn't been completed (e.g. in case of rebuffering) you can change the chunk that is currently downloading.

What does my ABR function outputs? (3)

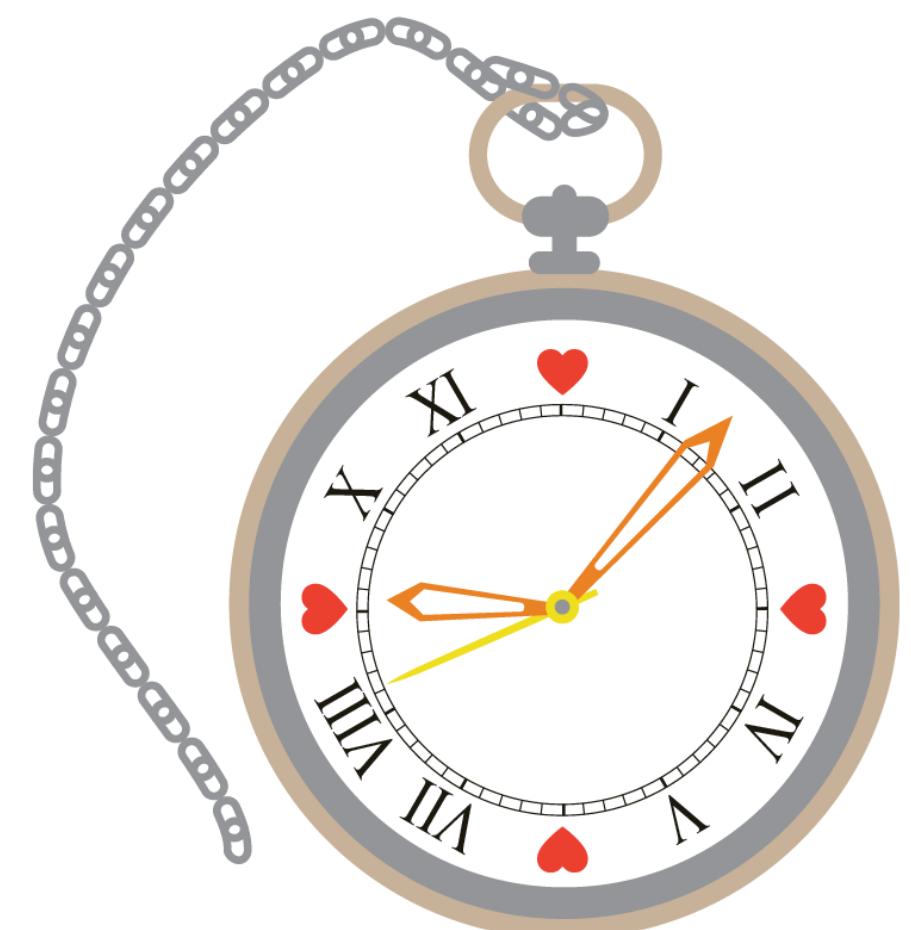
3. Timeout → set a timer that will trigger the ABR function again



What does my ABR function outputs? (3)

3. Timeout → set a timer that will trigger the ABR function again

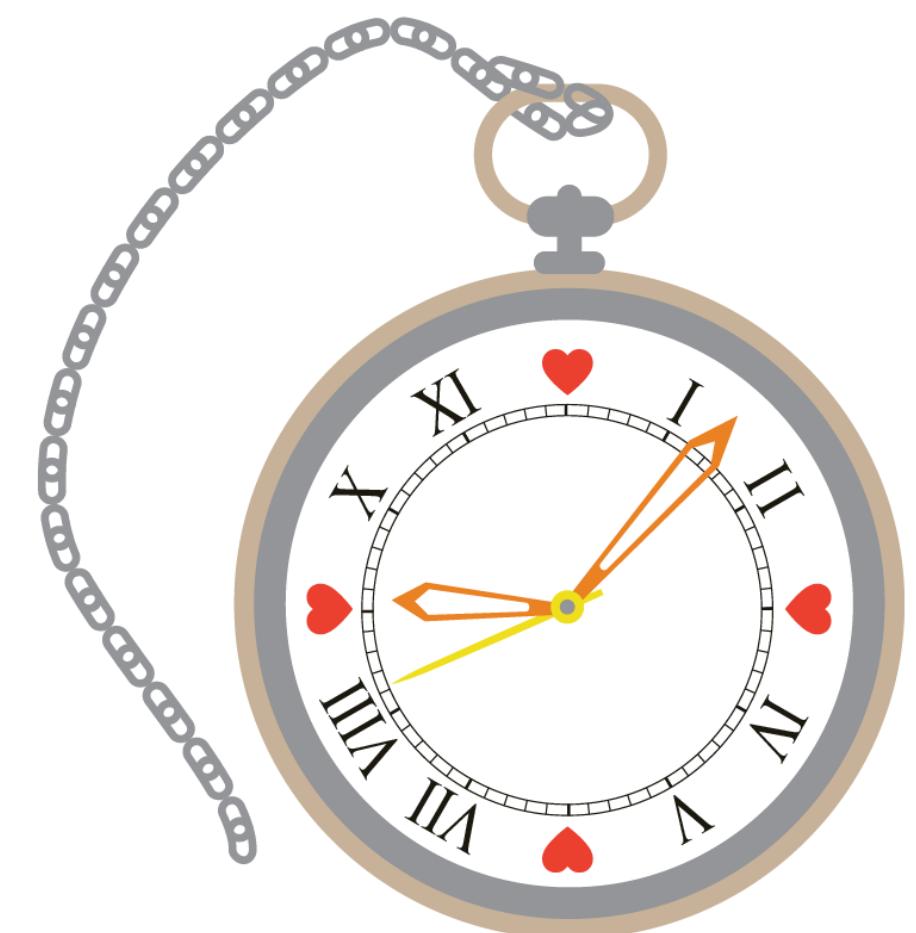
⌚ Timeout is in absolute time, usually set it as `current_time+X` (where min X is 200ms)



What does my ABR function outputs? (3)

3. Timeout → set a timer that will trigger the ABR function again

- ⌚ Timeout is in absolute time, usually set it as `current_time+X` (where min X is 200ms)
- ⌚ `Timeout == 0` means no timeout



ABR function inputs

- **Current time:** time from the beginning of the simulation (in seconds)
- Playback time: how much of the video has been shown (in seconds)
- Playback chunk: the chunk that is playing right now
- Current chunk: the chunk number that is downloading right now (or has been just finished)
- Current chunk quality: the quality of the current chunk
- Current chunk download: how much of current chunk has been downloaded (in bytes)
- Video: contains 6 video arrays (one per quality level). Each subarray contain the size of each chunk in the video

ABR function inputs

- Current time: time from the beginning of the simulation (in seconds)
- **Playback time:** how much of the video has been shown (in seconds)
- Playback chunk: the chunk that is playing right now
- Current chunk: the chunk number that is downloading right now (or has been just finished)
- Current chunk quality: the quality of the current chunk
- Current chunk download: how much of current chunk has been downloaded (in bytes)
- Video: contains 6 video arrays (one per quality level). Each subarray contain the size of each chunk in the video

ABR function inputs

- 🍄 Current time: time from the beginning of the simulation (in seconds)
- 🍄 Playback time: how much of the video has been shown (in seconds)
- 🍄 **Playback chunk**: the chunk that is playing right now
- 🍄 Current chunk: the chunk number that is downloading right now (or has been just finished)
- 🍄 Current chunk quality: the quality of the current chunk
- 🍄 Current chunk download: how much of current chunk has been downloaded (in bytes)
- 🍄 Video: contains 6 video arrays (one per quality level). Each subarray contain the size of each chunk in the video

ABR function inputs

- Current time: time from the beginning of the simulation (in seconds)
- Playback time: how much of the video has been shown (in seconds)
- Playback chunk: the chunk that is playing right now
- **Current chunk:** the chunk number that is downloading right now (or has been just downloaded)
- Current chunk quality: the quality of the current chunk
- Current chunk download: how much of current chunk has been downloaded (in bytes)
- Video: contains 6 video arrays (one per quality level). Each subarray contain the size of each chunk in the video

ABR function inputs

- Current time: time from the beginning of the simulation (in seconds)
- Playback time: how much of the video has been shown (in seconds)
- Playback chunk: the chunk that is playing right now
- Current chunk: the chunk number that is downloading right now (or has been just finished)
- **Current chunk quality:** the quality of the current chunk
- Current chunk download: how much of current chunk has been downloaded (in bytes)
- Video: contains 6 video arrays (one per quality level). Each subarray contain the size of each chunk in the video

ABR function inputs

- 🍄 Current time: time from the beginning of the simulation (in seconds)
- 🍄 Playback time: how much of the video has been shown (in seconds)
- 🍄 Playback chunk: the chunk that is playing right now
- 🍄 Current chunk: the chunk number that is downloading right now (or has been just finished)
- 🍄 Current chunk quality: the quality of the current chunk
- 🍄 **Current chunk download**: how much of current chunk has been downloaded (in bytes)
- 🍄 Video: contains 6 video arrays (one per quality level). Each subarray contain the size of each chunk in the video

ABR function inputs

- Current time: time from the beginning of the simulation (in seconds)
- Playback time: how much of the video has been shown (in seconds)
- Playback chunk: the chunk that is playing right now
- Current chunk: the chunk number that is downloading right now (or has been just finished)
- Current chunk quality: the quality of the current chunk
- Current chunk download: how much of current chunk has been downloaded (in bytes)
- **Video**: contains 6 video arrays (one per quality level). Each subarray contain the size of each chunk in the video

Task 1 & Task 2

Task 1 & Task 2

 428 bandwidth traces (*given*) → 428 experiments

Task 1 & Task 2

- 428 bandwidth traces (*given*) → 428 experiments
- A single video trace (*given*)



Task 1 & Task 2

- 428 bandwidth traces (*given*) → 428 experiments
- A single video trace (*given*)



● 1st threshold: Average score ≥ 75
4 points

Task 1 & Task 2

- 428 bandwidth traces (*given*) → 428 experiments
- A single video trace (*given*)

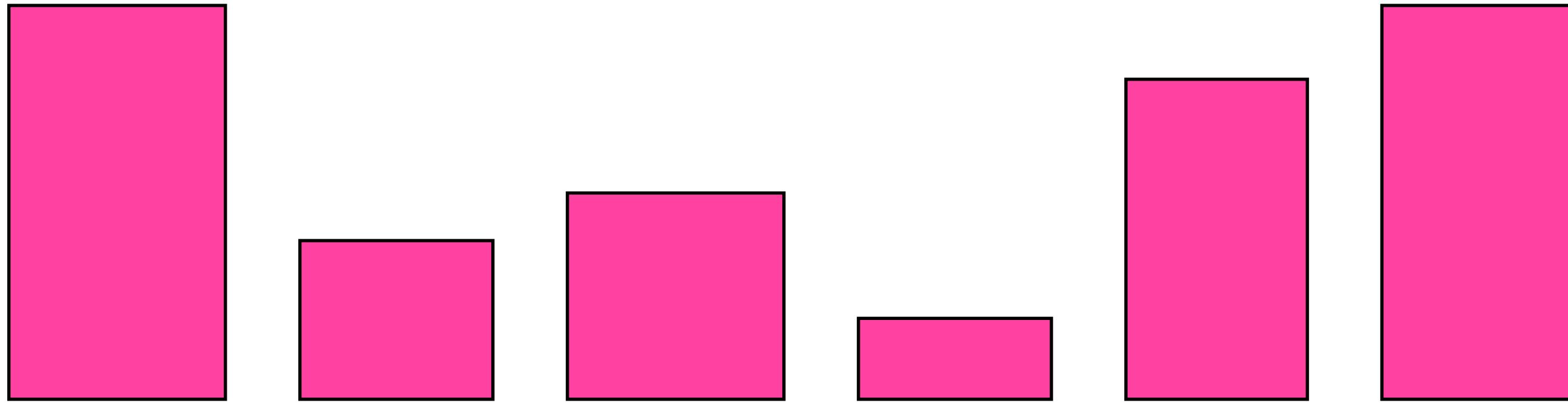


- 1st threshold: Average score ≥ 75
4 points
- 2st threshold: Average score ≥ 90
8 points

Before Task 3.. an observation

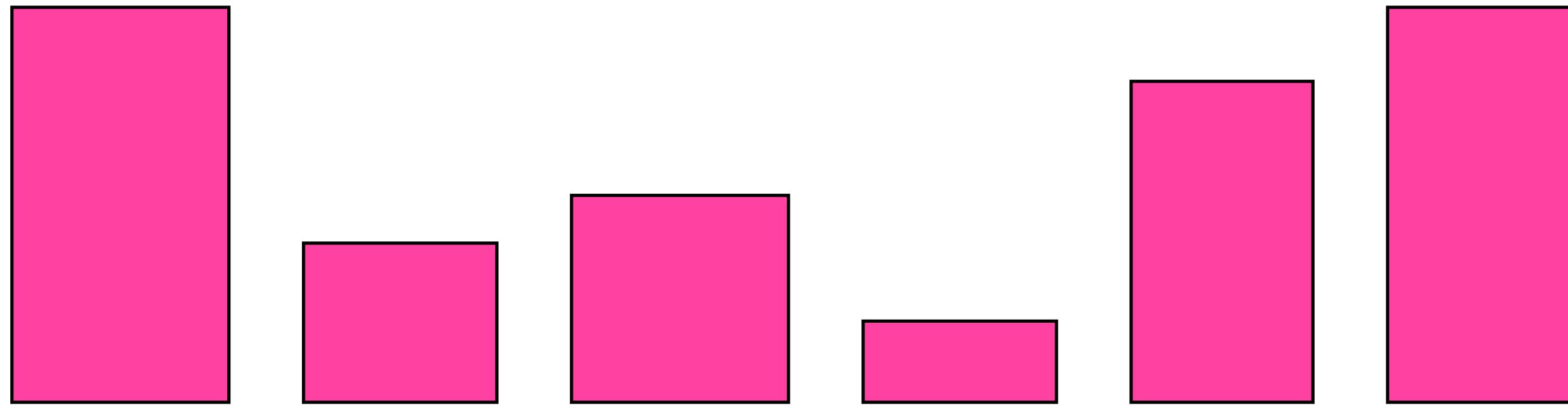
Before Task 3.. an observation

Video A
Res X

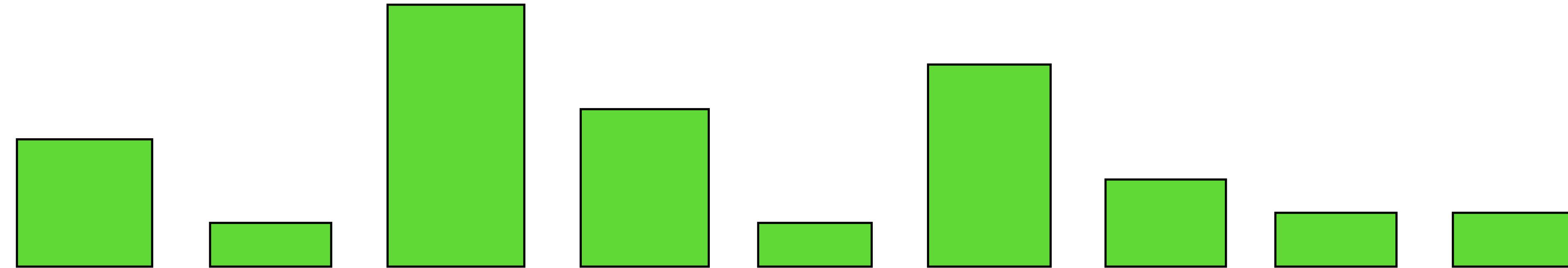


Before Task 3.. an observation

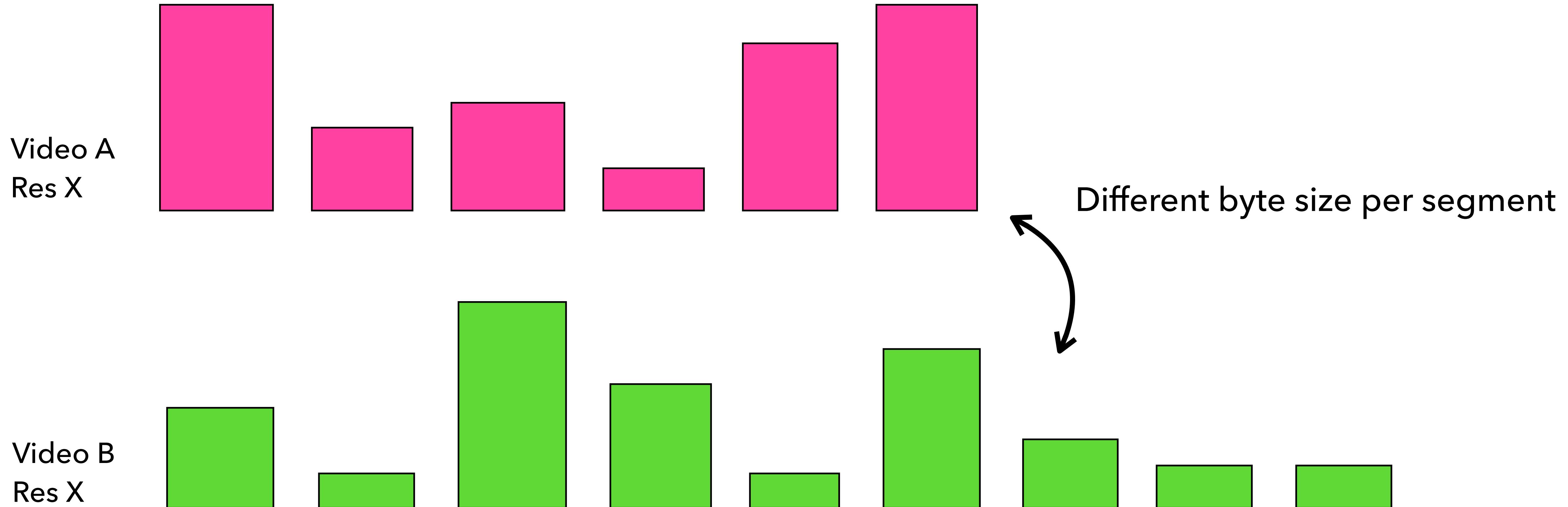
Video A
Res X



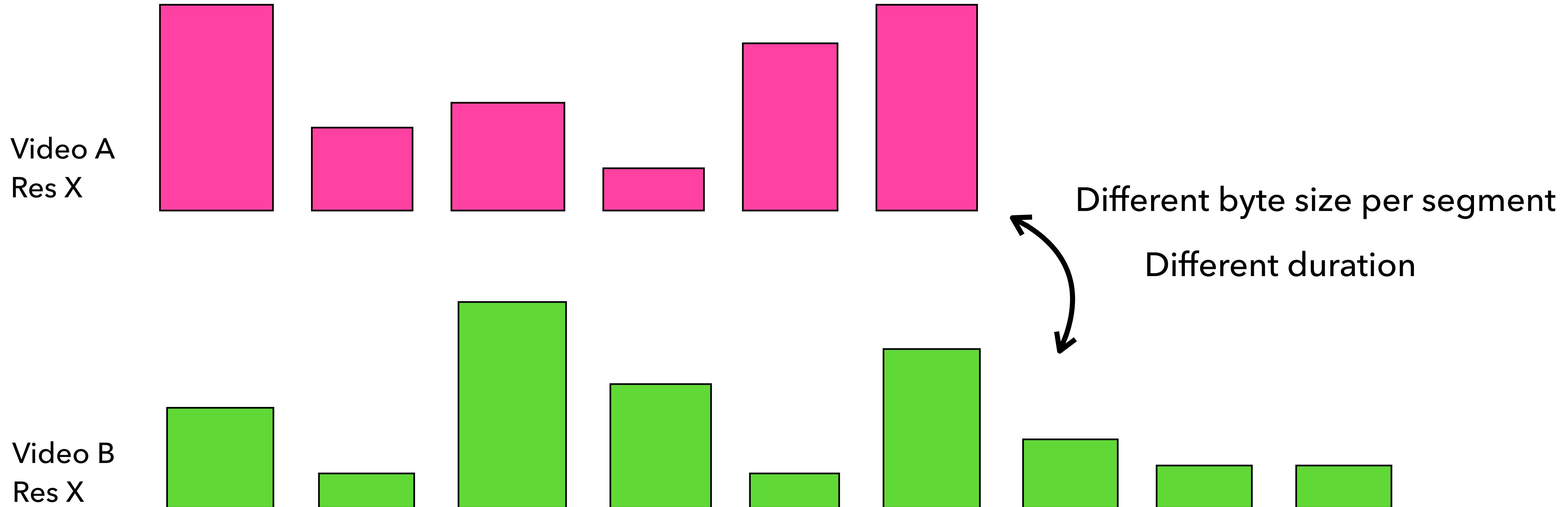
Video B
Res X



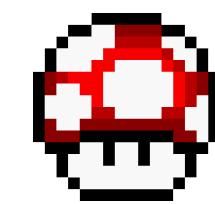
Before Task 3.. an observation



Before Task 3.. an observation



Task 3

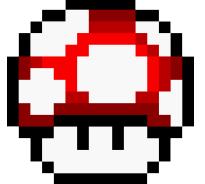
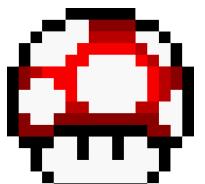
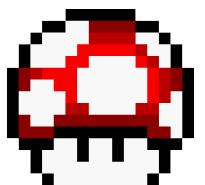


428 bandwidth traces (*given*)

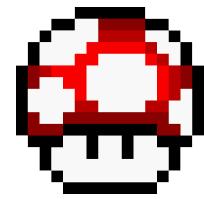
Task 3

-  428 bandwidth traces (*given*)
 -  10 video traces (*hidden*)
- ★ You have been provided with the script I used to generate videos (`generate_random_video.py`). Change the seed to obtain a different video trace to test your solution with!

Task 3

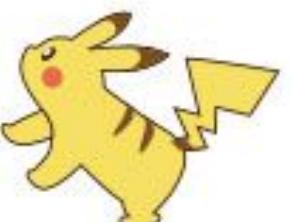
-  428 bandwidth traces (*given*)
-  10 video traces (*hidden*)
 - ★ You have been provided with the script I used to generate videos (`generate_random_video.py`). Change the seed to obtain a different video trace to test your solution with!
-  Task 3: get the best average QoE on the hidden video traces!

Legend

-  You are assigned two scores
- **Public score**: for Task 1 and Task 2, evaluated on the public video trace
 - **Hidden score**: for Task 3, evaluated on the hidden video traces. *The leaderboard is sorted by the hidden score.*

ABR contest

Threshold	Due	Text color	Poké Ball
75-	14.04. at 15:00	RED	
75 - 90	19.04. at 15:00	ORANGE	
90+	19.04. at 15:00	GREEN	
90+ and n.1	19.04. at 15:00	BLUE	



Reading the leaderboard

 If the leaderboard server is not able to run a solution, **ERROR** is displayed

Rank	Team Name	Public Score	Poké Ball	Hidden Score
1	Rihanna	110.081		110.922
2	Loathing in Vegas	109.734		109.672
3	Angie	108.401		107.846
4	Ankle Breakers	108.803		107.056
5	Master of Chunks	108.488		106.406

Reading the leaderboard

⌚ If the leaderboard server is not able to run a solution, **ERROR** is displayed

Rank	Team Name	Public Score	Poké Ball	Hidden Score
1	Rihanna	110.081		110.922
2	Loathing in Vegas	109.734		109.672
3	Angie	108.401		107.846
4	Ankle Breakers	108.803		107.056
5	Master of Chunks	108.488		106.406

Max points for Task 3

Reading the leaderboard

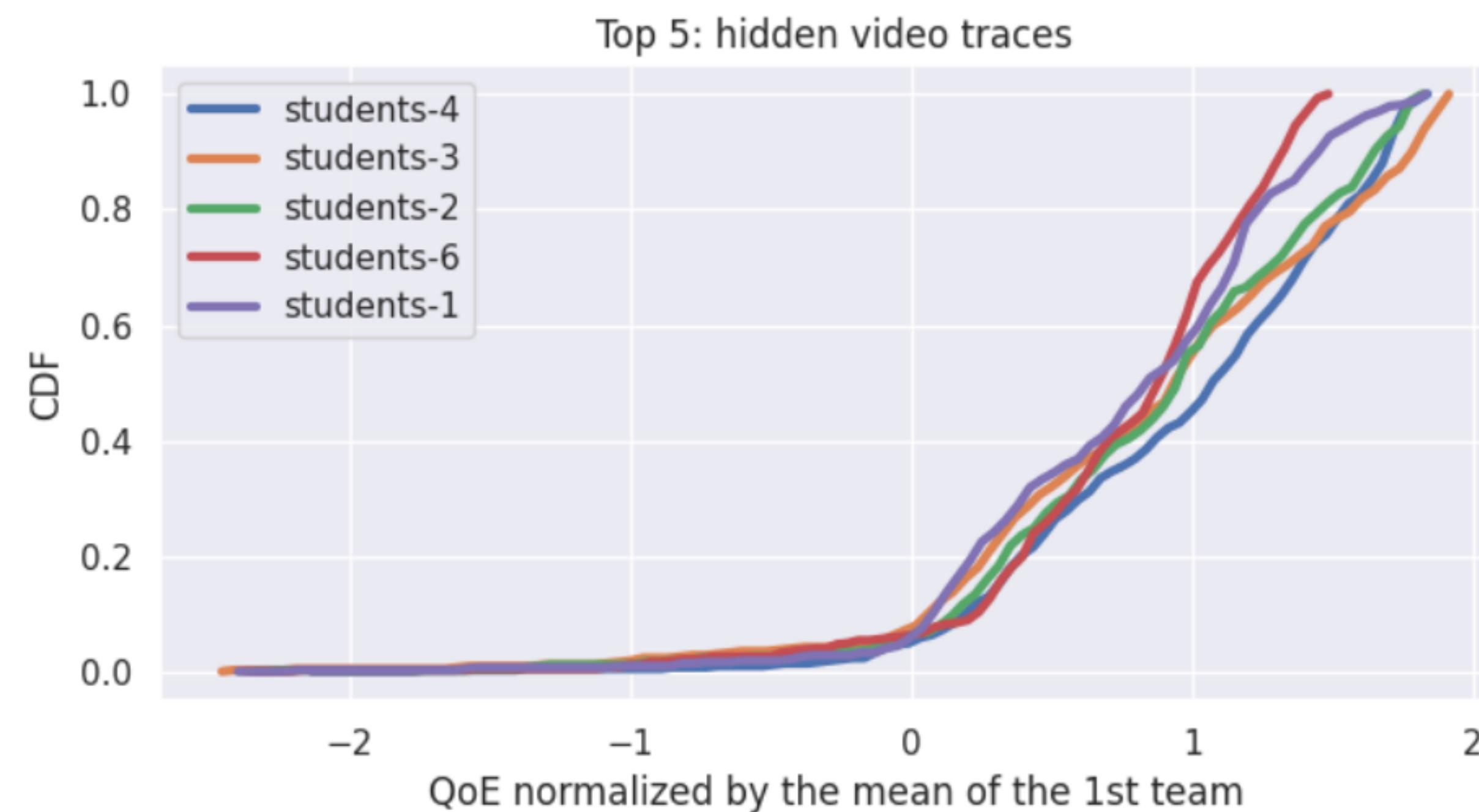
⌚ If the leaderboard server is not able to run a solution, **ERROR** is displayed

Rank	Team Name	Public Score	Poké Ball	Hidden Score
1	Rihanna	110.081		110.922
2	Loathing in Vegas	109.734		109.672
3	Angie	108.401		107.846
4	Ankle Breakers	108.803		107.056
5	Master of Chunks	108.488		106.406

Max points for Task 3

Linear interpolation
with a granularity of
0.25

For the best 5 teams, the CDF is shown!



The leaderboard is recalculated approximately every 2 minutes. The page should automatically refresh.

Last recalculation: 2020-03-24 10:05:10

Important dates

🍄 First Threshold deadline:

14/04/2020 15:00.

- If your public score is < 75, a penalty of 20% will be applied on the final mark.

🍄 Project deadline: 19/04/2020 15:00

🍄 Demo session: 20/04/2020 →

Remember to write the explain.MD!



READ.me

- Environment: Python 3.7
- Do not change the code in the simulation environment: I will plug your function in my simulation setup
- Do not try to hardcode traces (I'll go through your code anyway)
- Runtime limit: ~5 mins (just for the leaderboard)
- If your solution requires some additional python packages:
 - Write a setup.sh
 - Ping me, I'll install them in the server that is running the leaderboard evaluation

To conclude..

- Pixelated mushroom icon Try to prepare questions/clarifications for the Q&A session/first weeks of the project → no DDOS attack of questions/queries in the last week before the deadline

To conclude..

- Pixelated bomb icon Try to prepare questions/clarifications for the Q&A session/first weeks of the project → no DDOS attack of questions/queries in the last week before the deadline

Be polite and respectful!



Questions?