High performances clients for RabbitMQ

Gabriele Santomaggio

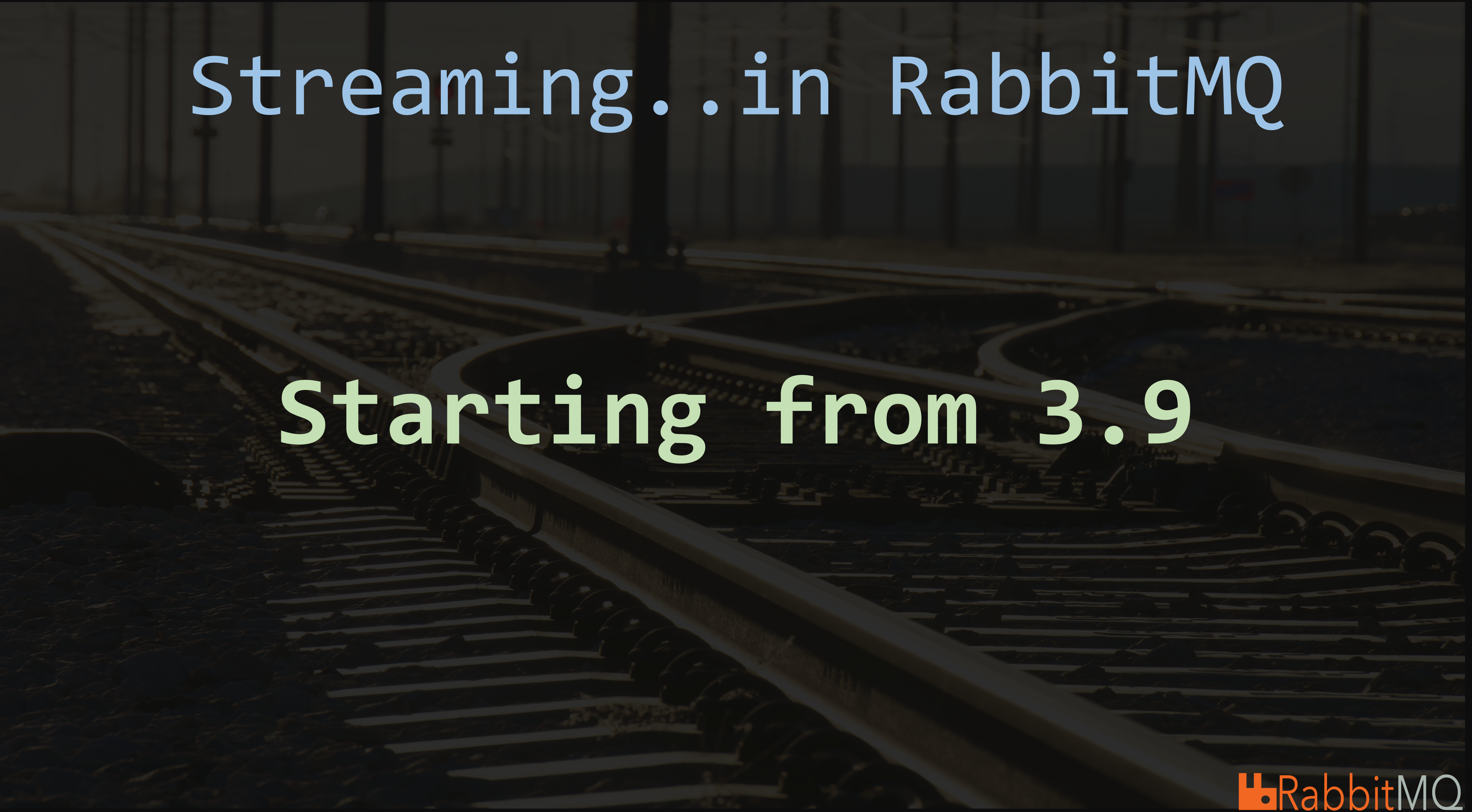
RabbitMQ Team

@gsantomaggio

vmware®

RabbitMQ

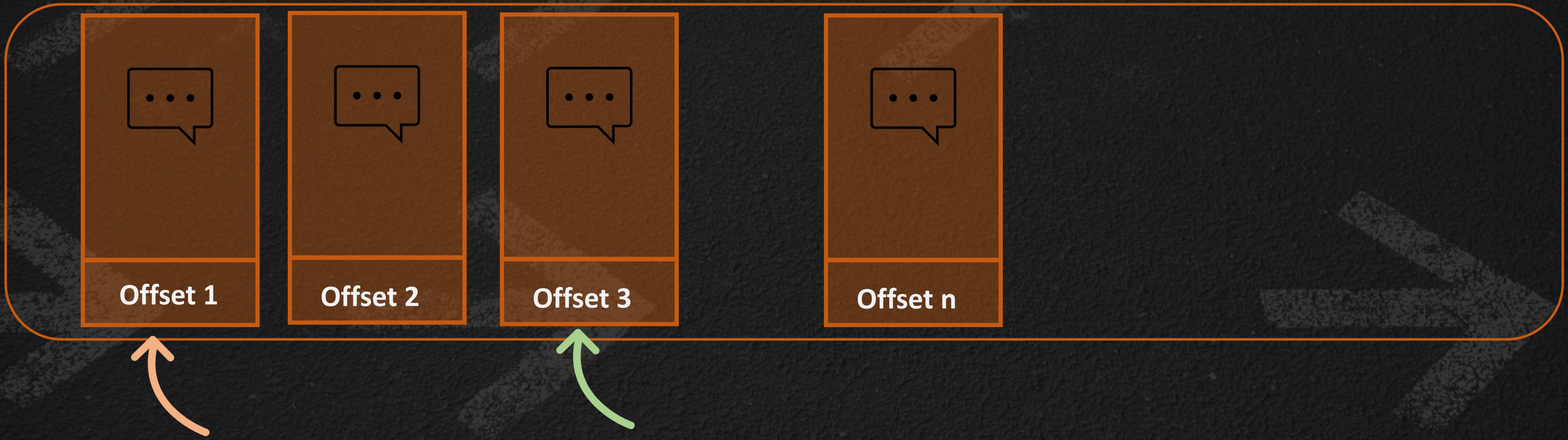
Streaming..in RabbitMQ



Starting from 3.9

RabbitMQ

Streaming..



Stream Performances

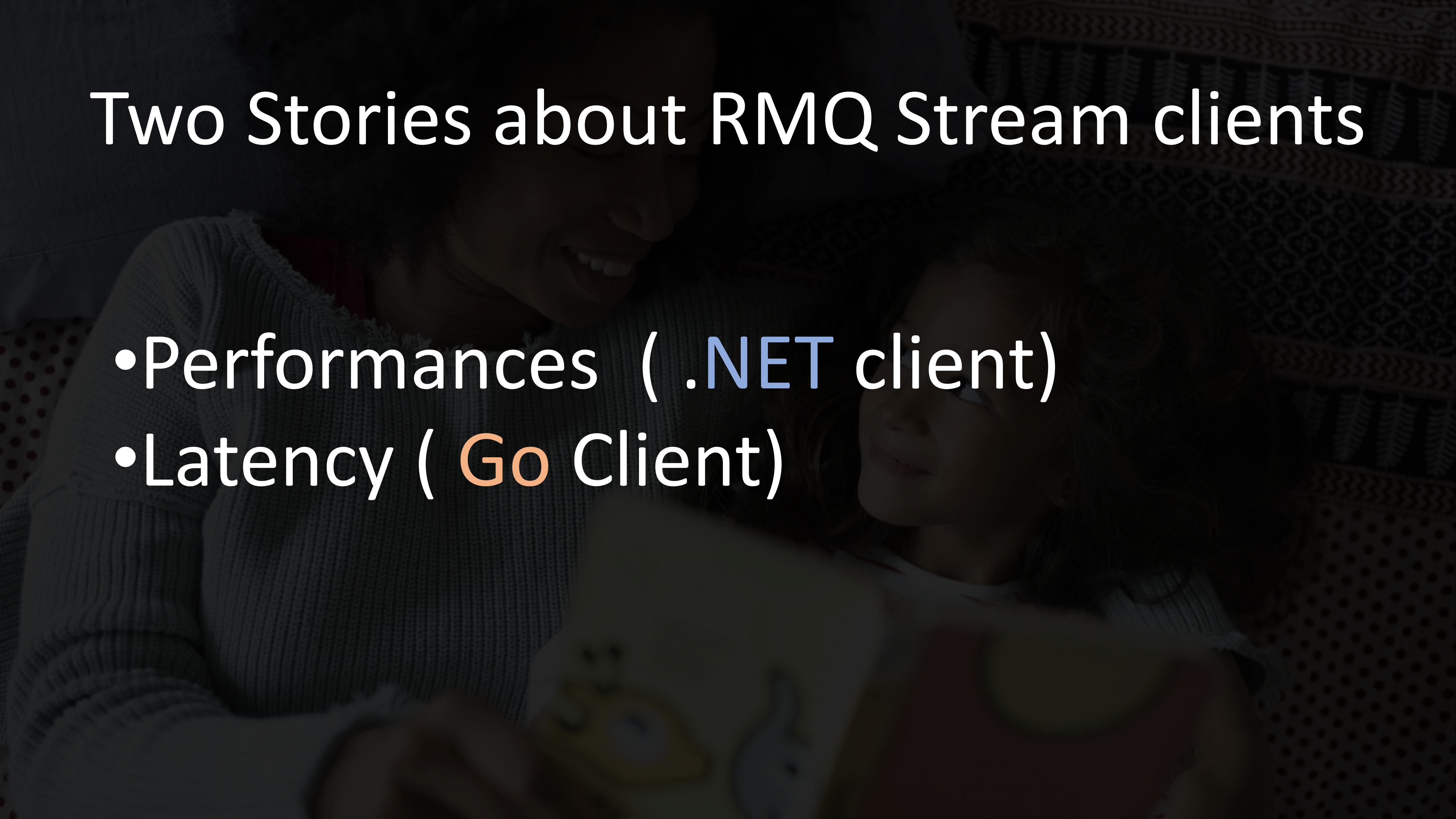
Millions messages per second

RabbitMQ

Sharing our experience

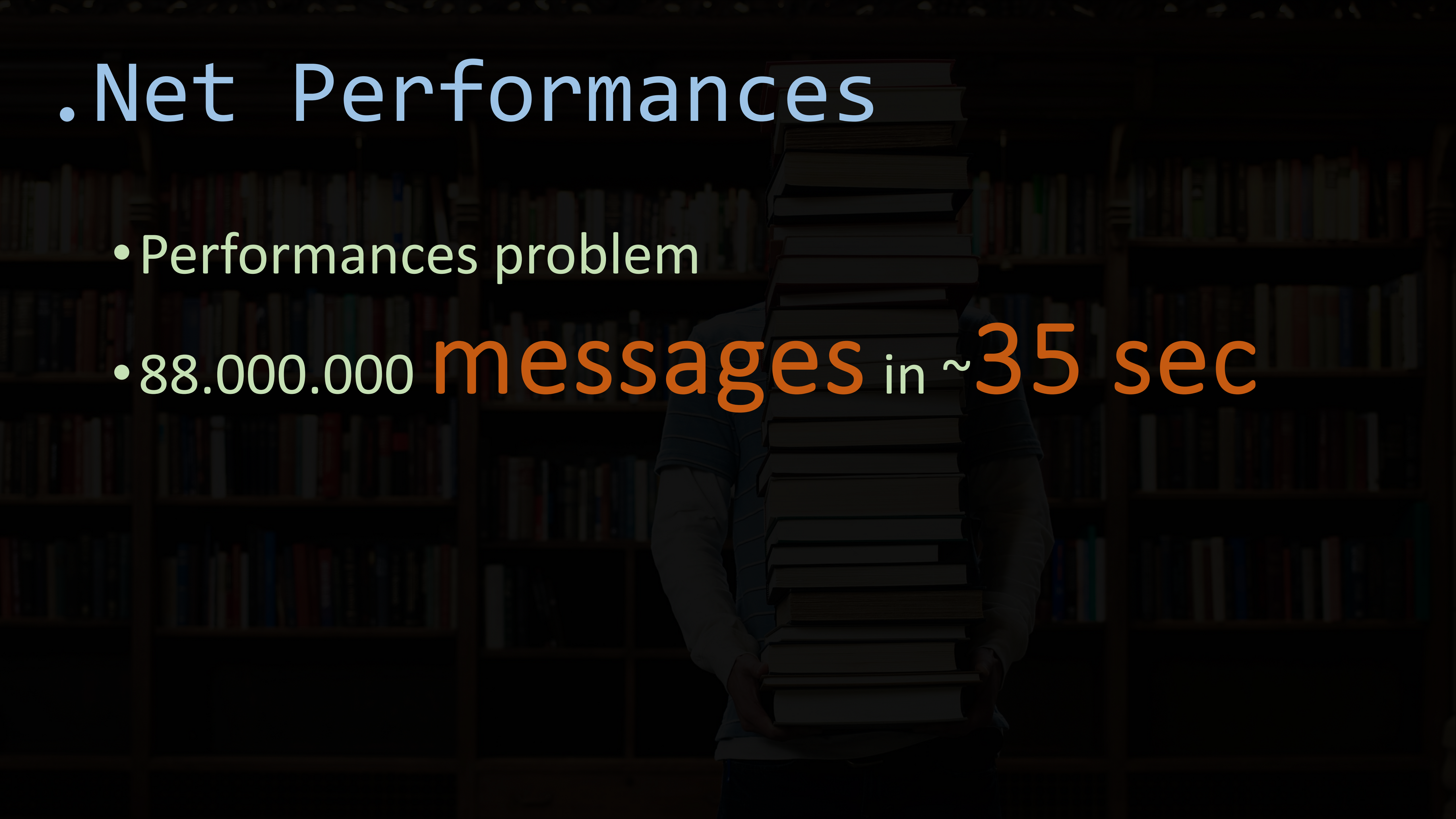
Two Stories about RMQ Stream clients

- Performances (.NET client)
- Latency (Go Client)



.NET

RabbitMQ

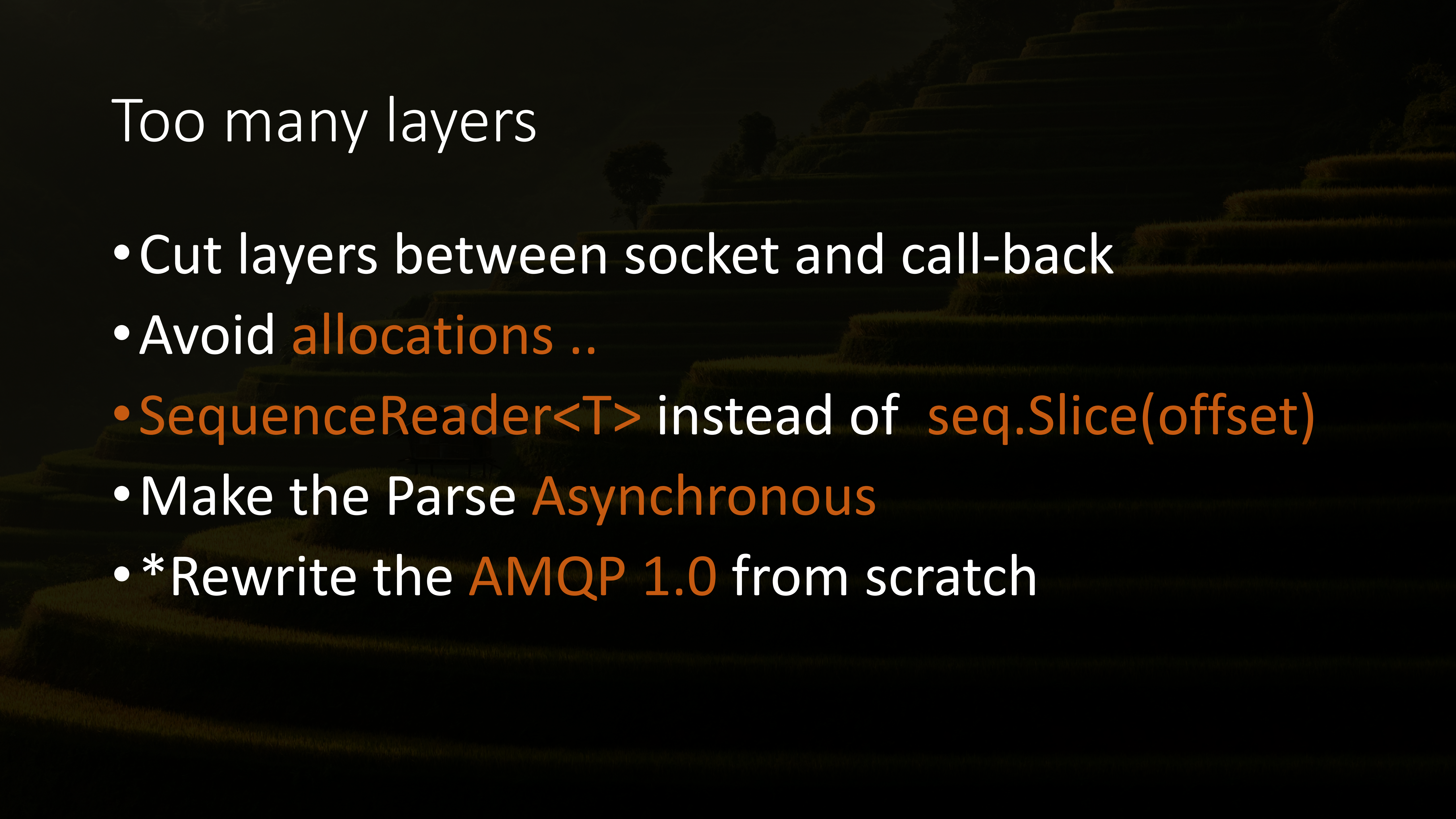


.Net Performances

- Performances problem
- 88.000.000 messages in ~35 sec

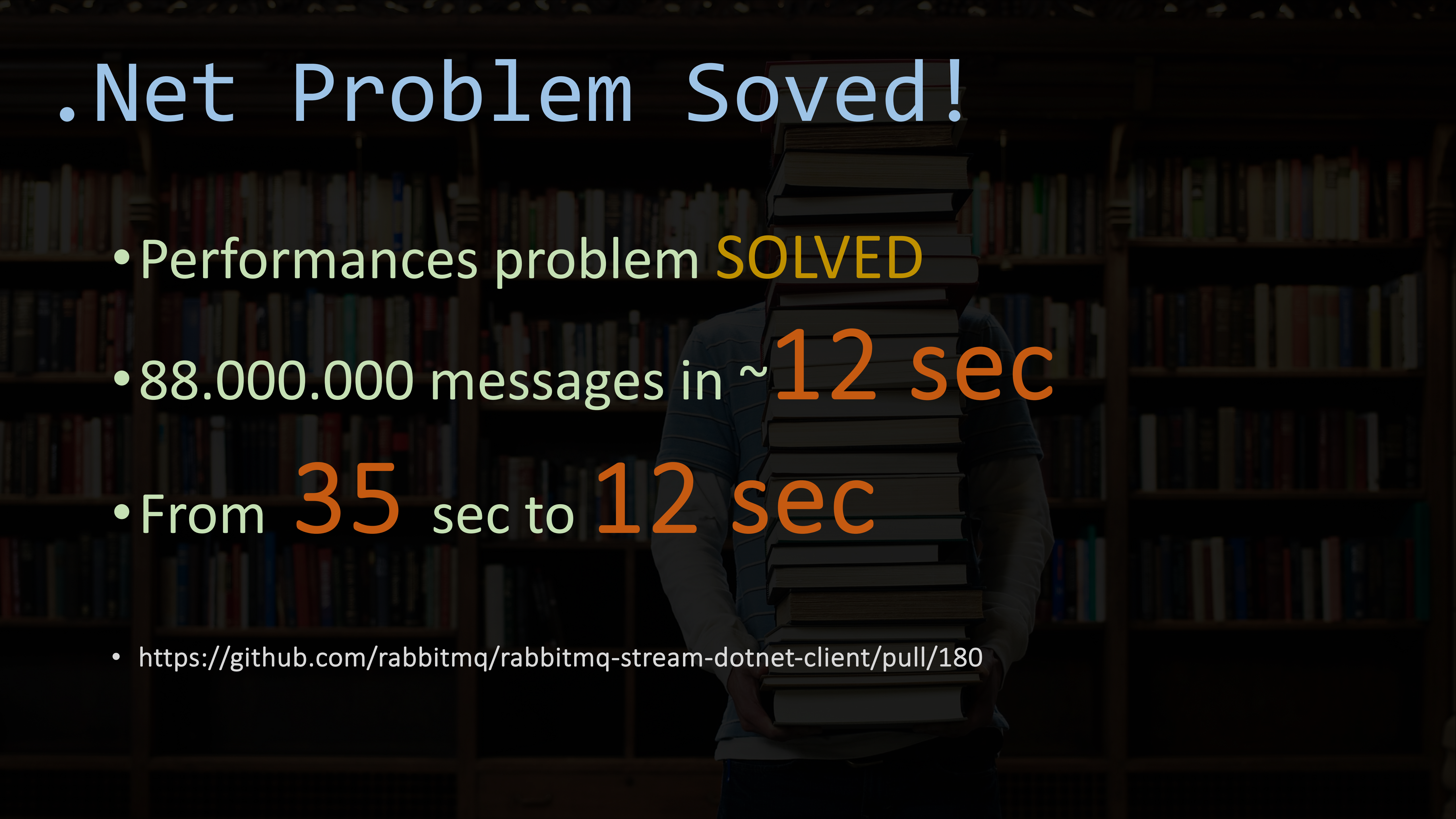
Solution?

Husqvarna
Husqvarna
562 XP



Too many layers

- Cut layers between socket and call-back
- Avoid allocations ..
- SequenceReader<T> instead of seq.Slice(offset)
- Make the Parse Asynchronous
- *Rewrite the AMQP 1.0 from scratch



.Net Problem Soved!

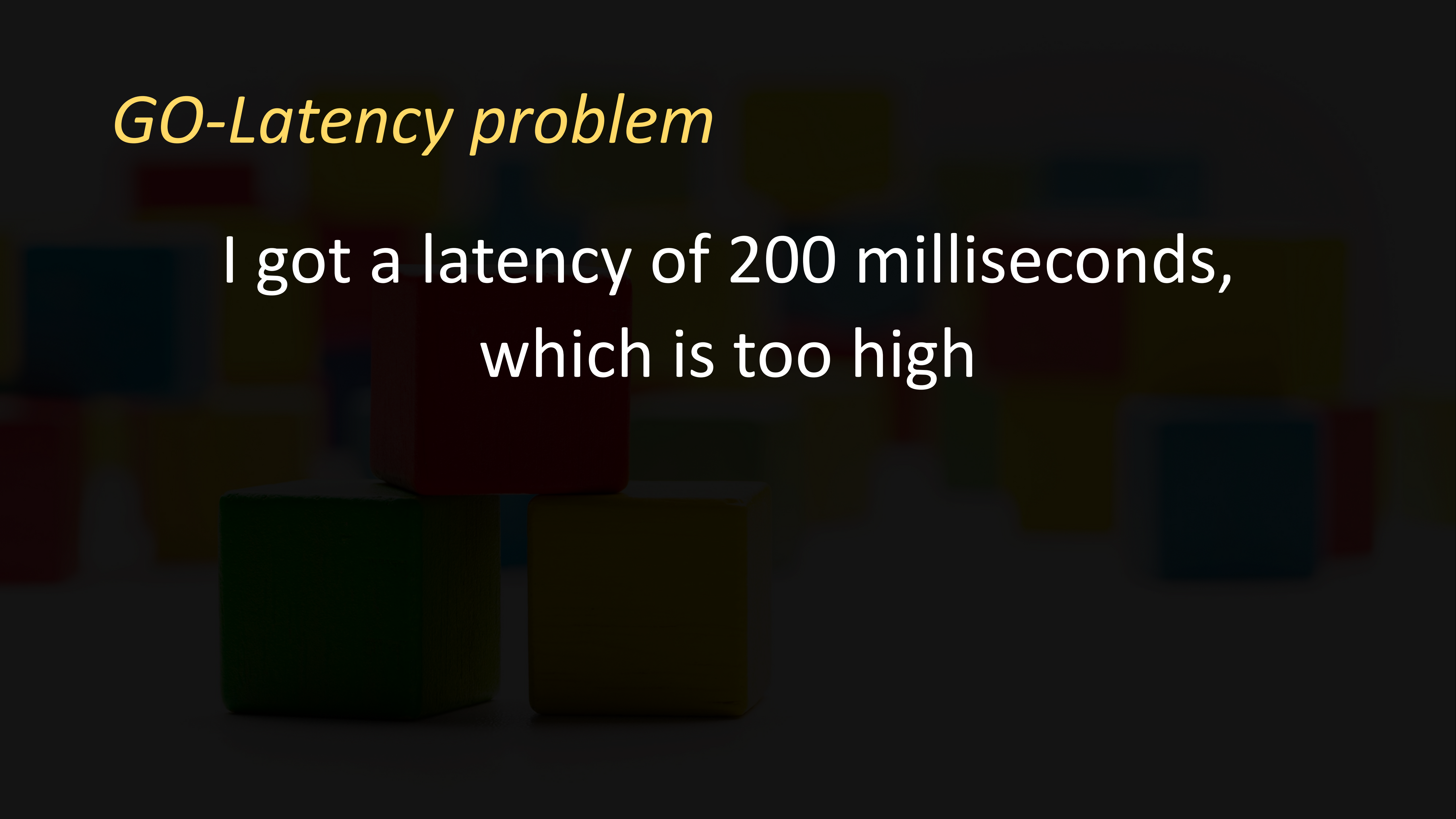
- Performances problem SOLVED
- 88.000.000 messages in ~12 sec
- From 35 sec to 12 sec

- https://github.com/rabbitmq/rabbitmq-stream-dotnet-client/pull/180

Golang

RabbitMQ

GO-Latency problem



I got a latency of 200 milliseconds, which is too high

Solution?

Optimization the write

- Writer Vs bufio.Writer
- `binary.Read(source, … , &res)` *// cool but slow*
- Provide a Sync low-level API to send the messages

GO-Latency problem- Solved

There is 100x improvement in results.

Elegant..

vs Fast

Ok let's conclude!

- Golang contains useful function in Sync.*
- .NET can be very fast (when you find the right way)
- Allocations are not free.. Recycle is better
- Serialization is expensive (ignored during the tests)
- Dirty *sometimes* is the faster way

Thank you

(I am around)

Telegram RabbitMQ:
https://t.me/RabbitMQ_ita

Youtube:
GabrieleSantomaggio

@gsantomaggio