

Working Title: Serverside rendering for realtime AR applications

Julius Neudecker
Bachelor of Science
julius.neudecker@haw-hamburg.de

Octobre 2020

Contents

1	Introduction	4
1.1	What is Serverside rendering?	4
1.2	State of technology - 4th quarter 2020	4
1.3	Technological challenges	4
1.3.1	Mobile device constraints	4
1.3.2	Network	4
1.3.3	Bandwidth	4
1.3.4	Packet loss	4
1.3.5	Server ressources	4
1.4	Deduction of Hypothesis	4
2	Architecture	4
2.1	Considerations	4
2.2	How to lightweight and fast	4
2.2.1	nVIDIA Maxine	4
2.3	Loadbalancing and Containerization	4
3	Implementation	4
3.1	Cross Platform vs. Ecosystems	4
3.2	Rapid changing Hard- and Software environments	4
3.3	nVIDIA cloud XR	4
4	Evaluation and research	4
4.1	Define research goals	4
4.2	User surveys	4
4.3	Technical analysis	5
5	Conclusion	5
5.1	Results	5
5.2	Future Work	5
5.3	Acknowledgements	5

This is a super fancy abstract, outlining the approach and findings of this masters thesis. More content t.b.d.

1 Introduction

1.1 What is Serverside rendering?

1.2 State of technology - 4th quarter 2020

1.3 Technological challenges

1.3.1 Mobile device constraints

1.3.2 Network

This is especially important for user experience.

- 5G Testsite at HH Harbour
- WIFI 6
- Software defined Network

1.3.3 Bandwidth

1.3.4 Packet loss

1.3.5 Server resources

1.4 Deduction of Hypothesis

Something along bc. of this and that, does it make sense from an UX point of view to have serverside rendering?

2 Architecture

2.1 Considerations

2.2 How to lightweight and fast

2.2.1 nVIDIA Maxine

Encode videostreams with ML to save up to 90

2.3 Loadbalancing and Containerization

3 Implementation

3.1 Cross Platform vs. Ecosystems

3.2 Rapid changing Hard- and Software environments

3.3 nVIDIA cloud XR

Serverside rendering platform for AR/VR Contents.

4 Evaluation and research

4.1 Define research goals

4.2 User surveys

In order to evaluate UX, HCI aspects have to be considered.

4.3 Technical analysis

5 Conclusion

5.1 Results

5.2 Future Work

5.3 Acknowledgements