

Prüfen:

- Liveness
- Safety
- Semantics

→ Once msg sent on one replica, eventually all replicas will have updated their msg
 Im report erwähnen wegen fancy? → All replicas go through safe sequence of msgs?

Requirements:

- Msg Counter pro Client und causal reihenfolge der Nachrichten

Append only Log ⇒ An Log immer nur angehängt, nie verändert

Jeder Client:

- Chat GUI
- Local CROT-Log
- Synchronisationslogik → Stand austauschen, obwohl nur hash / nummer etc.
& dann Fehlende aufzeigen
- Netzwerkstack

ClientList	
• statisches array mit # Clients gefüllt mit	▷ or alternative: each client stores its own seq # directly. All participants if client out of sync with other clients don't need to know seq # of others. Is there no diff seq?
Client Node	
• ClientNodes	- Client Name - Client Seq Num = # of Msg Sent
+ find SeqNumOfClient (name)	
+ update SeqNum (name, seqNum)	
+ wi Compose SeqNum (name, incomingNode)	
Lösbar und soll bei audio nichts msg haben so prüfen	
	→ others don't need to know seqNumbers, get Parents

Von C zu Arduino:

- struct ✓
- pointer ✓
- extern ✓
- function prototypes ✓
- header files gibts auch funktioniert leicht anders

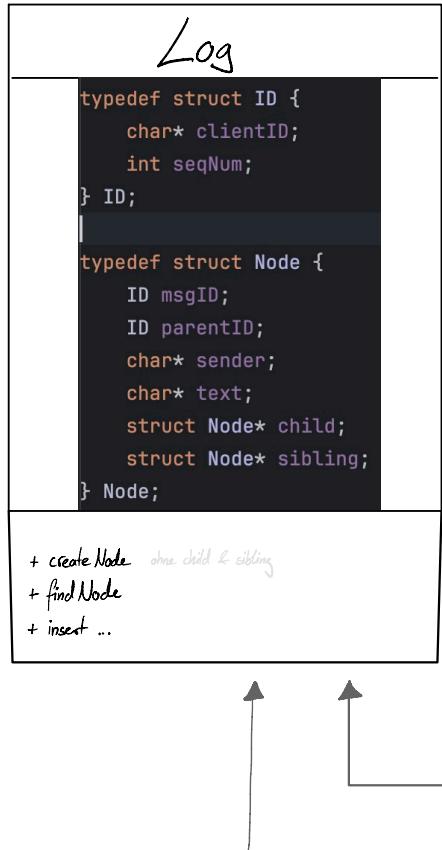
• kein Malloc / free! × → ?

• beschränkte Standard bibs (kein printf)
↳ Serial.print()

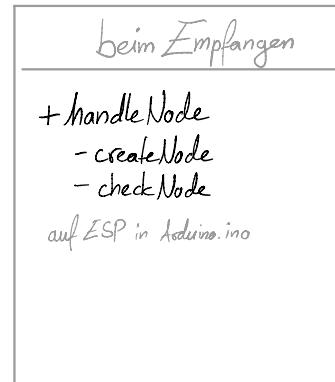
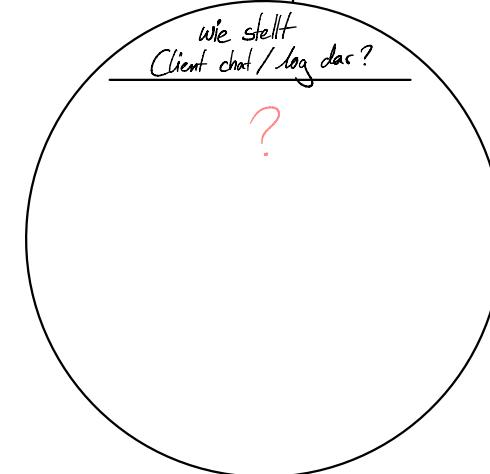
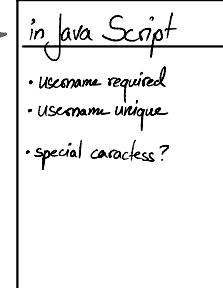
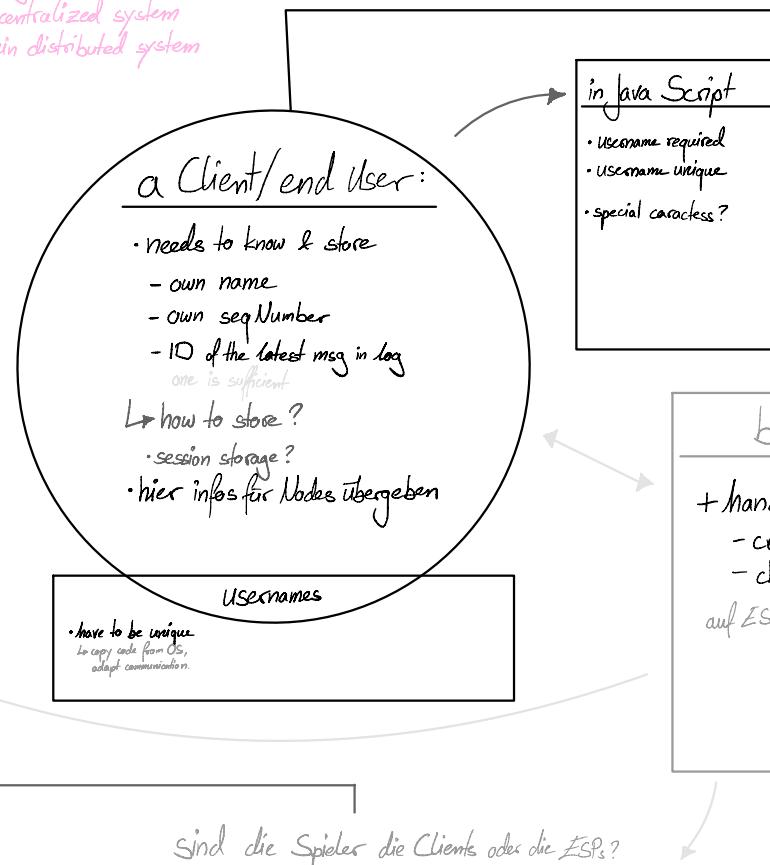
Aber:

• Logik in C möglich, einbinden mit [extern "C"]

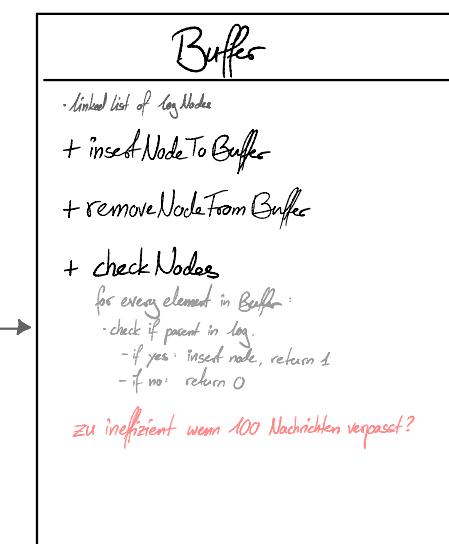
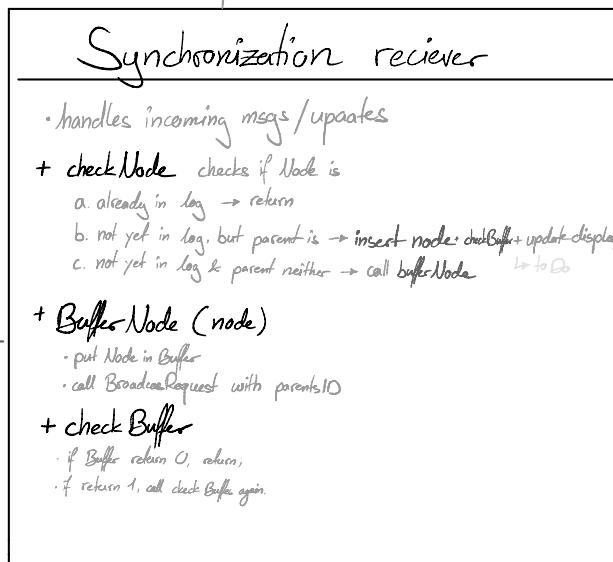
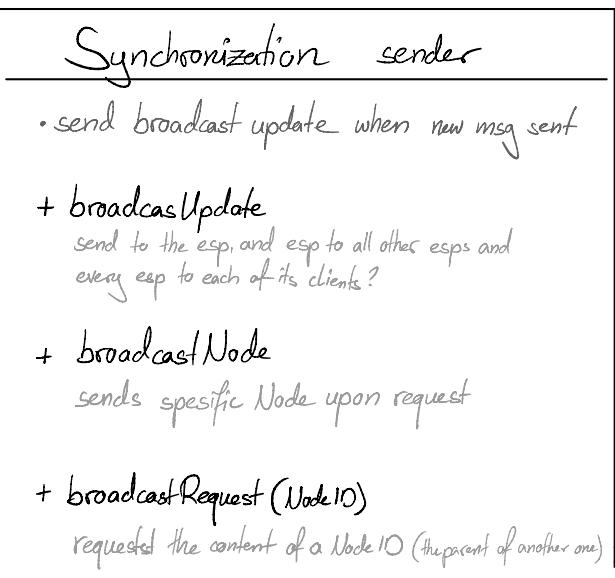
on each Esp



wenn logs nur auf ESP
=> decentralized system
und kein distributed system



- Unterscheidung nötig zwischen Client und esp!
- was wenn App anstatt Webseite?
 - ↳ hätten wir dann einfach eigene tiny SSB entwickelt?
 - ↳ ihn fragen



What happens when one gets disconnected?

- ESP / log : secondary storage?
- To user:
 - username ? } if one persistent
 - seqNumber ? } other has to be too
 - parent (or just use last com msg / seqNum - 1 as parent)