

CREATING VIRTUAL WORLDS (2)

ICE

V. Jurenka

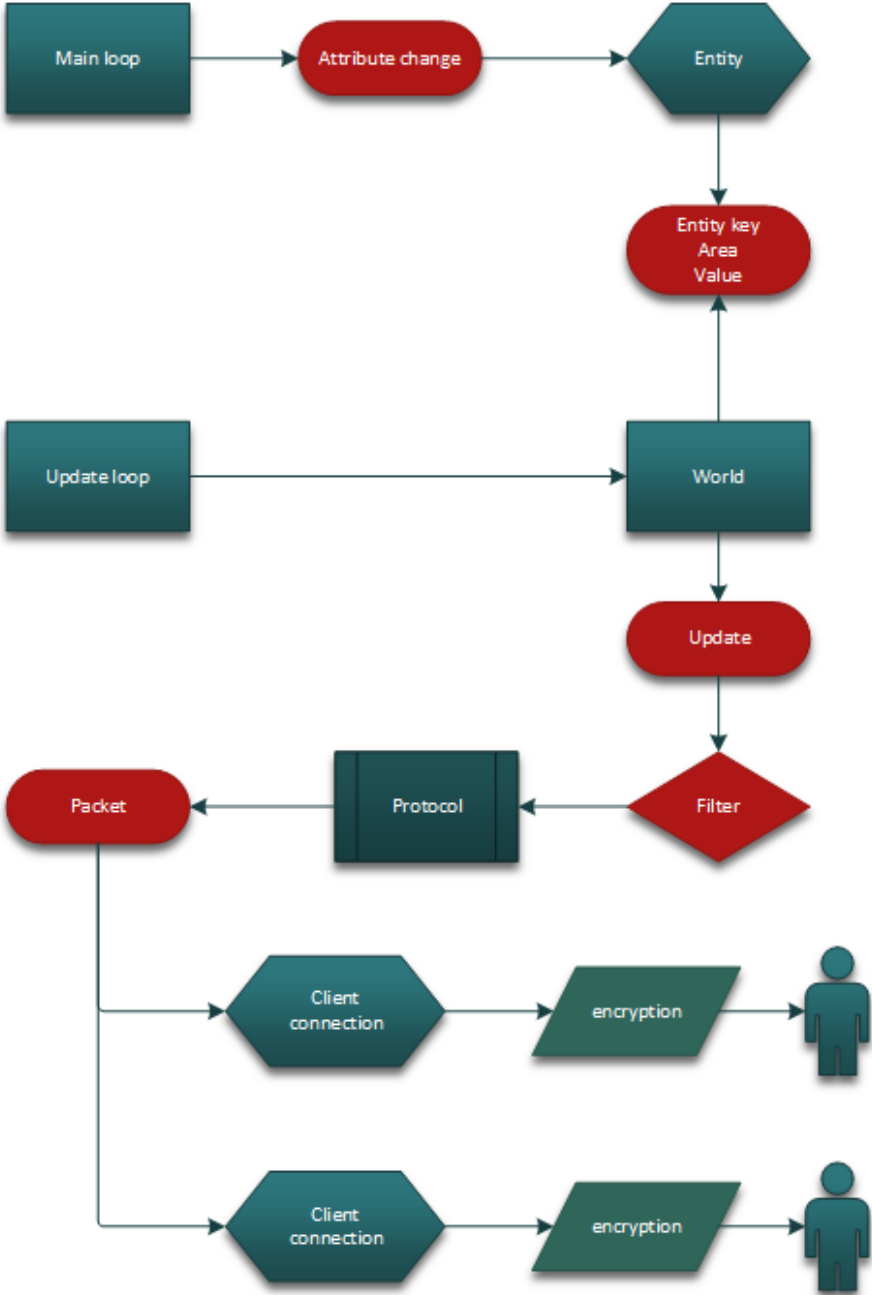
Hibernate

- Direct mapping of sql tables to classes and vice versa
- Not enough -> dynamic schema for entity stats
- Caching everywhere

Data layer

- 1) create a template, store the most important attributes in table in db (name, icon, model name) – needed often by both server and client
- 2) define common rendering attributes, save them in cache and xml – only client needs
- 3) spawn *base entity* in editor, store in xml - only server needs
- 4) server first initializes from base entities in xml, then stores them in tables – this allows resets

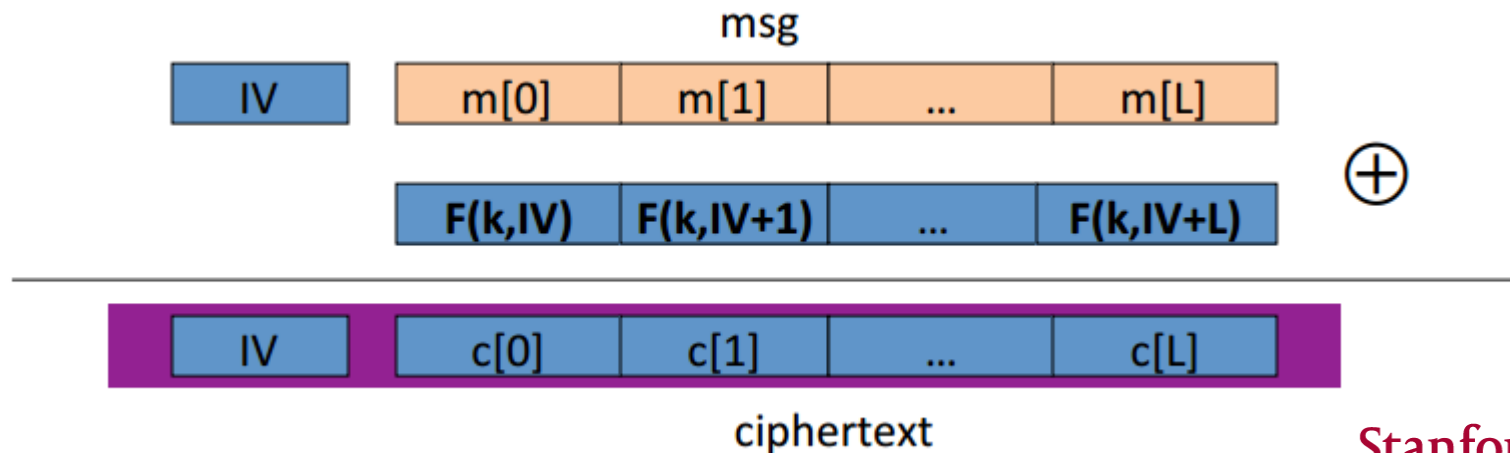
Server logic



Network

- RSA for login and exchange of AES key

AES – CTR mode



- + means XOR
- $F(k, m)$ -> encrypt message 'm' using AES with key 'k'
- IV – initialisation vector, 128 bit -> 64 bits nonce + 64 bits counter
- Must not reuse pair key, IV -> leads to two times pad

Example packet processing

```
0000000000000000000000000000000011010000101011010110100111100101011111110000101100010  
10000100110010101101011111110010000010100011101000100110100010001101010101100000  
11010101010010000101010100000011010011111101101010101010101001011111101000011110
```

00000000 00000000 00000000 00011010 | 000101011010110100111100101011111110000101100...

26 bytes follow (will change int to short later)

decrypt using AES in CTR mode

```
00110000 00000000 00000001 00101110 11000001 00000000 00000000 00001010 00110011 00000010 00000000 00000000 00000000  
00001010 01000000 00000000 00000000 00000000 01000000 00000000 00000000 00000000 01000000 00000000 00000000 00000000
```

let the protocol parse

```
00110000 | 00000000 00000001 00101110 11000001 | 00000000 00000000 00001010 00110011 | 00000010 |  
00000000 00000000 00000000 00001010 | 01000000 00000000 00000000 00000000 |  
01000000 00000000 00000000 00000000 | 01000000 00000000 00000000 00000000
```

packet_type | target_class_hash | target_id | attribute_type | map_id | x | y | z

48 -> entity_update | class "npc" | 2611 | 2 -> position | 10 | 2 | 2 | 2