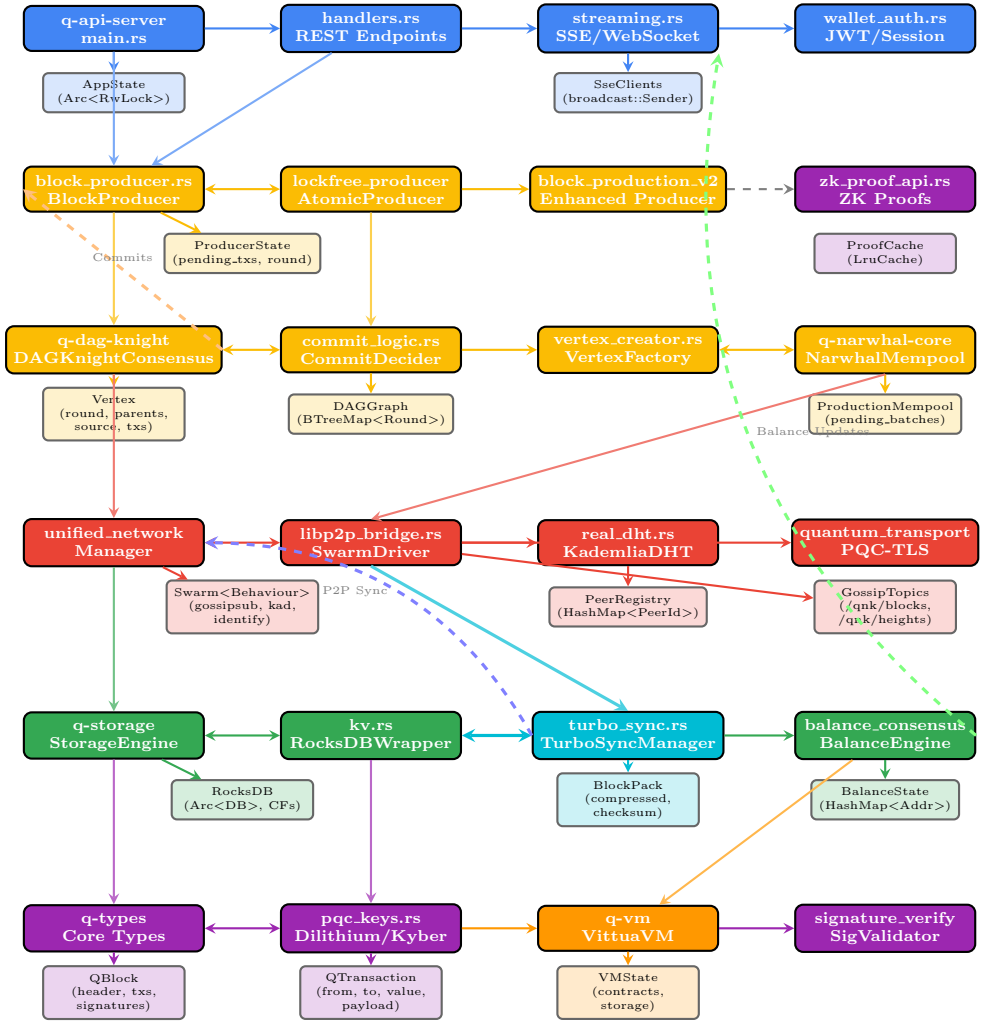
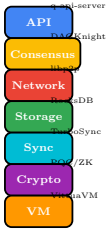
Q-NarwhalKnight Architecture Diagram

Complete Module Communication Map





Key Data Flows:

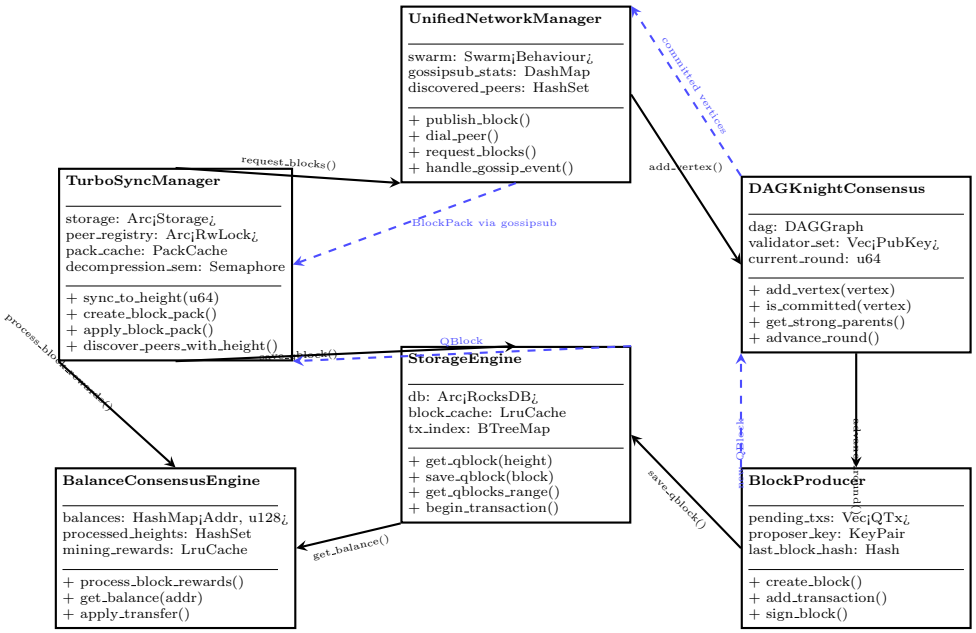
P2P Sync: TurboSync ↔ Network
Compressed BlockPacks via gossipsub

Balance Updates: Storage → SSE
Real-time balance via Server-Sent Events

Commits: DAGKnight → Producer
Finalized vertices trigger new blocks

Class Communication Details

Method Calls and Message Flows



Sync Sequence:

1. TurboSync.discover_peers_with_height()
2. Network.request_blocks(peer, range)
3. Peer creates BlockPack via create_block_pack()
4. Gossipsub delivers compressed pack
5. TurboSync.apply_block_pack()
6. LZ4 decompress → postcard/bincode
7. Storage.save_qblock() for each block
8. BalanceEngine.process_block_rewards()

Block Production:

1. DAGKnight.is_committed() → true
2. BlockProducer.create_block()
3. Vertex created with parents
4. Network.publish_block()
5. Gossipsub propagates to peers
6. Peers add to their DAG

Decentralization Architecture Evidence

Code-Level Analysis

1. No Leader Election (Fully Decentralized)
Any validator can propose blocks at any time

```
// vertex_creator.rs
pub async fn create_vertex(&self) -> Vertex {
   // No leader check - any node can create
   Vertex { source: self.validator_id, ... }
}
```

2. DAG Parallel Consensus
Multiple blocks can be created simultaneously

```
// commit_logic.rs
fn get_strong_parents() -> Vec<Hash> {
   // Reference MULTIPLE parent vertices
   self.dag.get_tips().collect()
}
```

3. Gossipsub Propagation
No central relay - mesh network topology

```
// unified_network_manager.rs
gossipsub.publish(topic, message)
// Propagates to random mesh peers
// No single point of control
```

4. Kademlia DHT Discovery
Decentralized peer discovery

```
// real_dht.rs
kad.get_closest_peers(peer_id)
// Distributed hash table lookup
// No central directory server
```

5. Byzantine Fault Tolerance
Tolerates malicious minority

```
// commit_logic.rs
const BFT_THRESHOLD: f64 = 2.0 / 3.0;
// Requires 2/3 honest validators
// No single entity can control
```

Testnet Concerns (Not Protocol Issues)
- Bootstrap nodes are known (but DHT reduces reliance)
- Small validator set (permissionless joining planned)

Expert Verdict: The protocol is decentralized. Current operational centralization (bootstrap nodes, small validator set) are testnet constraints, not architectural limitations.