Technology Stack Research

Research Real-time Communication Protocols (WebSocket, gRPC):

A. WebSocket (Full Duplex Real-time Communication)

A full-duplex communication channel between agent and CMS.

Why WebSocket can help

- Real-time push of policies
- Live alerts
- Reduced polling overhead
- Event-driven architecture

Pros

- Simpler implementation
- Native in browsers (good for dashboards)

Cons

- Less structured than gRPC
- Binary protocols need custom handling
- Harder to secure compared to TLS+REST

How WebSocket Works

- Starts as an HTTP connection
- Upgrades to a persistent two-way channel
- Server can push data anytime
- Client communicates without polling

Does WebSocket Work for Your EDR? → YES (Optional for Real-Time Alerts)

Why It Works

✓ Ideal for pushing urgent alerts (malware found, suspicious activity)
✓ Good for real-time dashboards
✓ Lower latency than HTTP polling
✓ Simple libraries available in Python + C++

Limitations

✗ Harder to secure than synchronous HTTPS
✗ Not ideal for large bulk telemetry
✗ Needs keep-alive + reconnection logic

Best Use Case

- Real-time dashboard
- Live policy updates
- Instant threat alerts

Suitability Score: 7/10

→ Good if you want live monitoring from dashboard.

B. gRPC (Google Remote Procedure Call[RPC Framework])

High-performance, strongly typed communication protocol using Protocol Buffers.

Why gRPC is ideal for EDR

- Very efficient (uses HTTP/2)
- Built-in streaming (server → agent → server)
- Strong typing via .proto files
- Automatic code generation
- Easier to implement real-time telemetry pipelines

Pros

- Fast & scalable
- Great for agent–server communication

- Secure via Built-in TLS support
- Lower bandwidth usage

Cons

- More complex than REST
- Requires protobuf schema design

How gRPC Works

- Uses HTTP/2
- Uses Protocol Buffers (binary format)
- Communication defined in .proto files(As mentioned before)
- Supports:
    - Unary request/response
    - Server streaming
    - Client streaming
    - Bidirectional streaming

Does gRPC Work for Your EDR? → YES (Ideal for large-scale systems)

Why It Works

✓ Extremely fast → ideal for high-frequency telemetry

✓ Smaller size than JSON → Lower bandwidth usage
✓ Strong typing prevents data mismatch
✓ Streaming allows continuous data flow

Limitations
✕ Debugging harder (binary payloads)

Suitability Score:

Prototype: **7/10**
Production scaling: **10/10**

→ Best for a future advanced version of BESS.