Automotive Digital Cockpit - Design Document

Author: Abhishek Chauhan

List of Figures

k, sockets.emit(_message_), and serverSideEmit() actor Developer as Dev participant _Socket_n(Cli

Figure 2: Cluster Communication Flow (Copy).png

Figure 3: Cluster Communication Flow.png

Figure 4: Frontend WS ping_pong + Backend Socket.IO serverSideEmit(_ping_) with ACKs(1).png

Figure 5: Frontend WS ping_pong + Backend Socket.IO serverSideEmit(_ping_) with ACKs(2).png

Figure 6: Frontend WS ping_pong + Backend Socket.IO serverSideEmit(_ping_) with ACKs.png

Figure 7: Frontend WebSocket ping_pong + Backend Socket.IO serverSideEmit(_ping_).png

Figure 8: Socket.IO Event Flow_ emit(), sockets.emit(_message_), and serverSideEmit().png

Figure 9: Socket.IO Event Flow_ emit(), write(_message_), and serverSideEmit().drawio.png

Figure 10: Socket.IO Flow_ emit() with _registerAckCallback, write(), and serverSideEmit().png

Figure 11: Socket.IO Flow_ emit(), sockets.emit(_message_), serverSideEmit().png

Figure 12: Socket.IO Flow_ emit(), write(_message_), and serverSideEmit().png

Figure 13: Socket.IO Sequence_ emit(), sockets.emit(_message_), serverSideEmit().png

Figure 14: Socket.IO client emit, server broadcast, and server-side emit.png

15: Socket.IO event flow with RESERVED_EVENTS check, write(_message_), and serverSideEmit.dra

e 16: Socket.IO flow_ client emit(), server sockets.emit(_message_), and io.serverSideEmit(...).drawi

e 17: Socket.IO flow_ emit() override (safe), write(_message_) broadcast, serverSideEmit() with ACK

Figure 18: Socket.IO flow_ emit() override, write(_message_), serverSideEmit().png

Figure 19: Socket.IO flow_ emit(), write(_message_), and serverSideEmit().png

Figure 20: Socket.IO flow_ emit_Ev_(), _registerAckCallback, write(_message_), serverSideEmit(.png

ure 21: Socket.IO flow_ safe emit(), namespace write(_message_), serverSideEmit() with ACKs(1).p

igure 22: Socket.IO flow_ safe emit(), namespace write(_message_), serverSideEmit() with ACKs.pn

Figure 23: Socket.IO flow_ send(), emit(), write(_message_), and serverSideEmit().png

Figure 24: Socket.IO flow_ socket.send_emit(), io.write(_message_), and io.serverSideEmit(...).png

Figure 25: Socket.IO lifecycle_ emit(), namespace broadcast, serverSideEmit().png

Figure 26: Socket.IO send()_emit(), write(_message_), and serverSideEmit().png

Figure 27: Socket.IO sequence_ emit(), write(_message_), serverSideEmit().png

Figure 28: Socket.IO sequence_ send(), emit(), write(_message_), and serverSideEmit().png

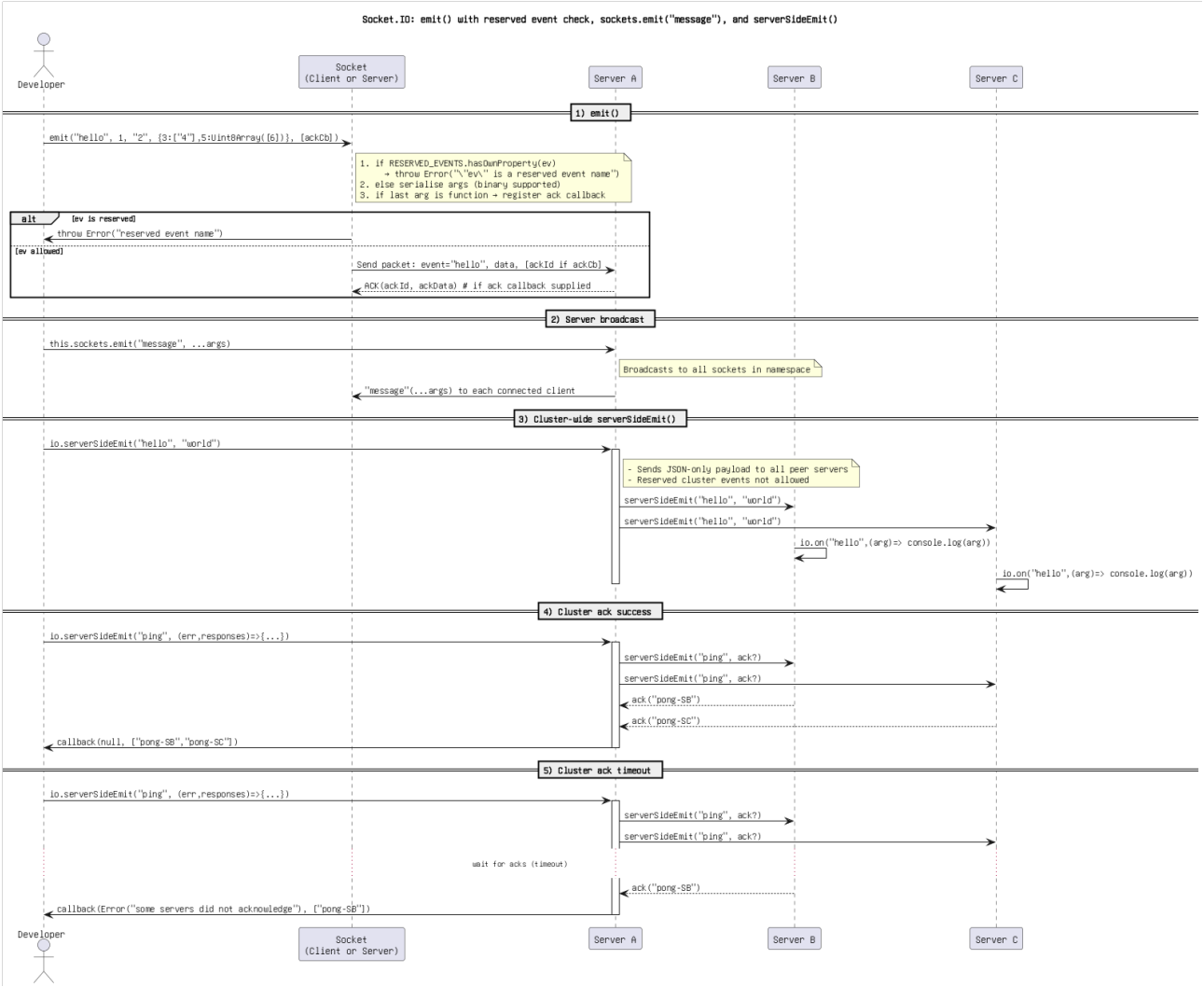
Figure 29: Socket.IO_ client emit(), server write(_message_), and io.serverSideEmit().png

Figure 30: Socket.IO_ emit() + _registerAckCallback + write(_message_) + serverSideEmit().png

re 31: Socket.IO_ emit() override + namespace _message_ broadcast + serverSideEmit() with ACKs

Figure 32: Socket.IO_ emit() override + namespace _message_ broadcast + serverSideEmit().png

gure 33: Socket.IO_ emit() override + namespace write(_message_) + serverSideEmit() with ACKs.p
Figure 34: Socket.IO_ emit() override + write(_message_) + serverSideEmit() with ACKs(1).png
Figure 35: Socket.IO_ emit() override + write(_message_) + serverSideEmit() with ACKs(2).png
Figure 36: Socket.IO_ emit() override + write(_message_) + serverSideEmit() with ACKs(3).png
gure 37: Socket.IO_ emit() override + write(_message_) + serverSideEmit() with ACKs(4) (Copy).pn
Figure 38: Socket.IO_ emit() override + write(_message_) + serverSideEmit() with ACKs(4).png
Figure 39: Socket.IO_ emit() override + write(_message_) + serverSideEmit() with ACKs..png
Figure 40: Socket.IO_ emit() override + write(_message_) + serverSideEmit() with ACKs.png
Figure 41: Socket.IO_ emit() override + write(_message_) + serverSideEmit()(1).png
Figure 42: Socket.IO_ emit() override + write(_message_) + serverSideEmit()(2).png
Figure 43: Socket.IO_ emit() override + write(_message_) + serverSideEmit()(3).png
Figure 44: Socket.IO_ emit() override + write(_message_) + serverSideEmit().png
Figure 45: Socket.IO_ emit() override, namespace write(_message_), cluster serverSideEmit().png
Figure 46: Socket.IO_ emit() override, namespace write(_message_), serverSideEmit().png
Figure 47: Socket.IO_ emit() override, write(_message_), serverSideEmit() with ACKs.png
Figure 48: Socket.IO_ emit() override, write(_message_), serverSideEmit().png
Figure 49: Socket.IO_ emit() with _registerAckCallback, write(_message_), serverSideEmit().png
Figure 50: Socket.IO_ emit(), write(_message_), serverSideEmit().png
Figure 51: Socket.IO_ send(), emit() override, write(_message_), serverSideEmit().png
Figure 52: Socket.IO_ send(), emit(), write(_message_), and serverSideEmit().png
Figure 53: Socket.IO_ send(), emit(), write(_message_), serverSideEmit().png
Figure 54: Socket.IO_ send()_write() via s.unshift + emit(), and serverSideEmit().png
gure 55: Untitled DiSocket.IO_ send(), emit() override, write(_message_), serverSideEmit()agram.pn
Figure 56: WebSocket ping_pong (frontend) vs Socket.IO serverSideEmit(_ping_) (backend).png
57: socket.IO Flow_ emit() with _registerAckCallback, sockets.emit(_message_), and serverSideEmi
ure 58: socket.IO Flow_ emit() with reserved event check, write(_message_), and serverSideEmit().p



k, sockets.emit(_message_), and serverSideEmit() actor Developer as Dev participant _Socket_n(Cli

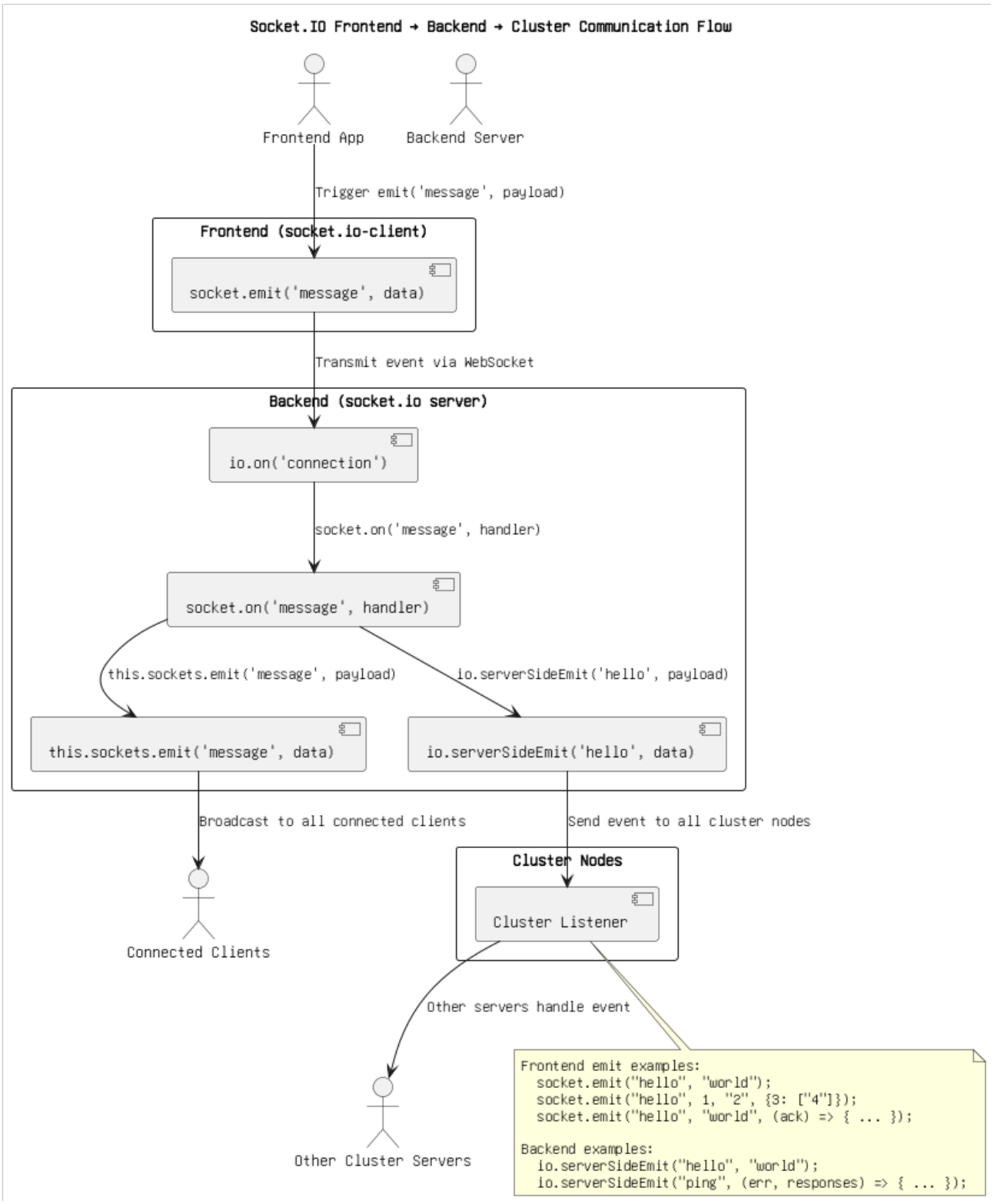


Figure 2: Cluster Communication Flow (Copy).png

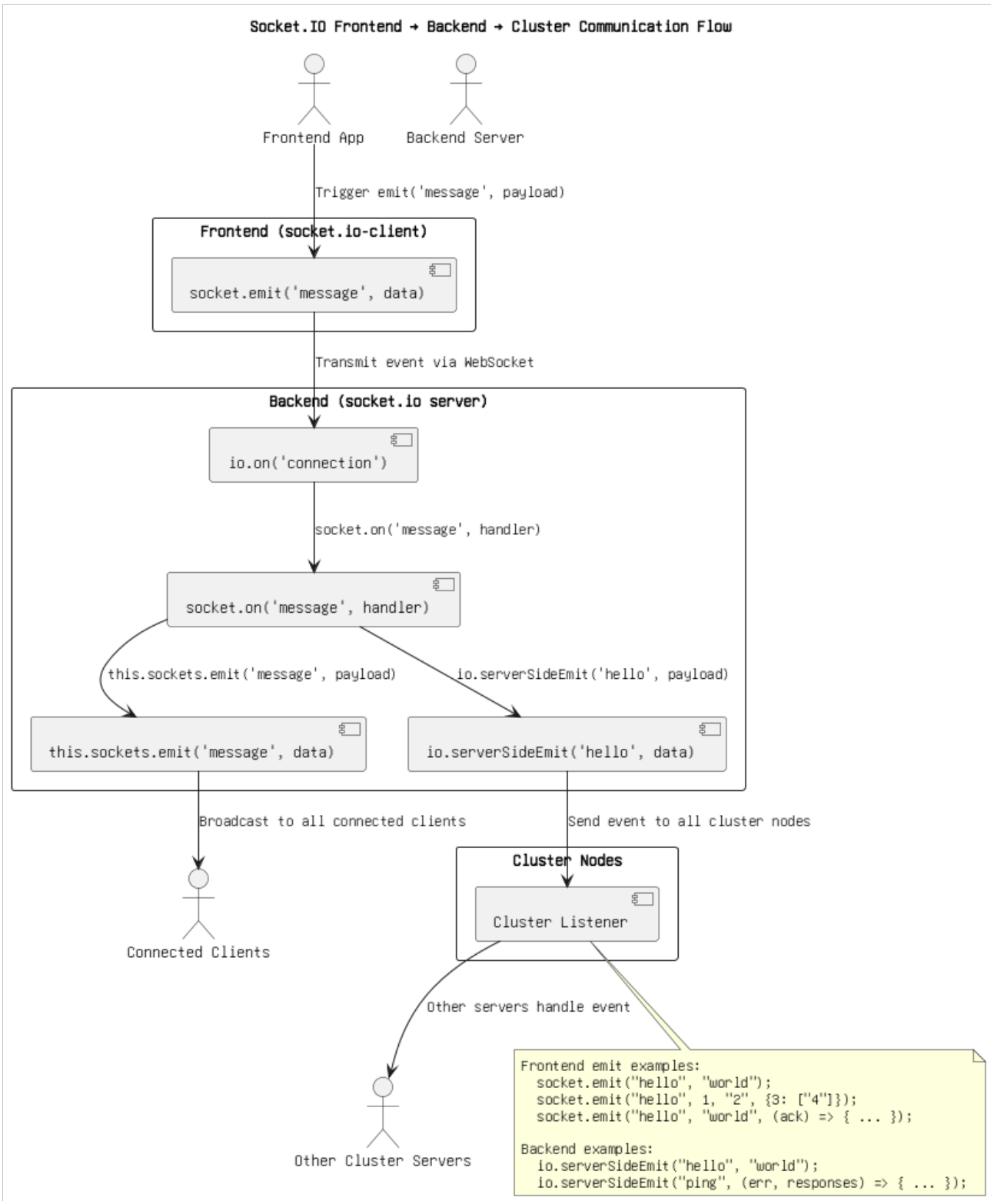


Figure 3: Cluster Communication Flow.png

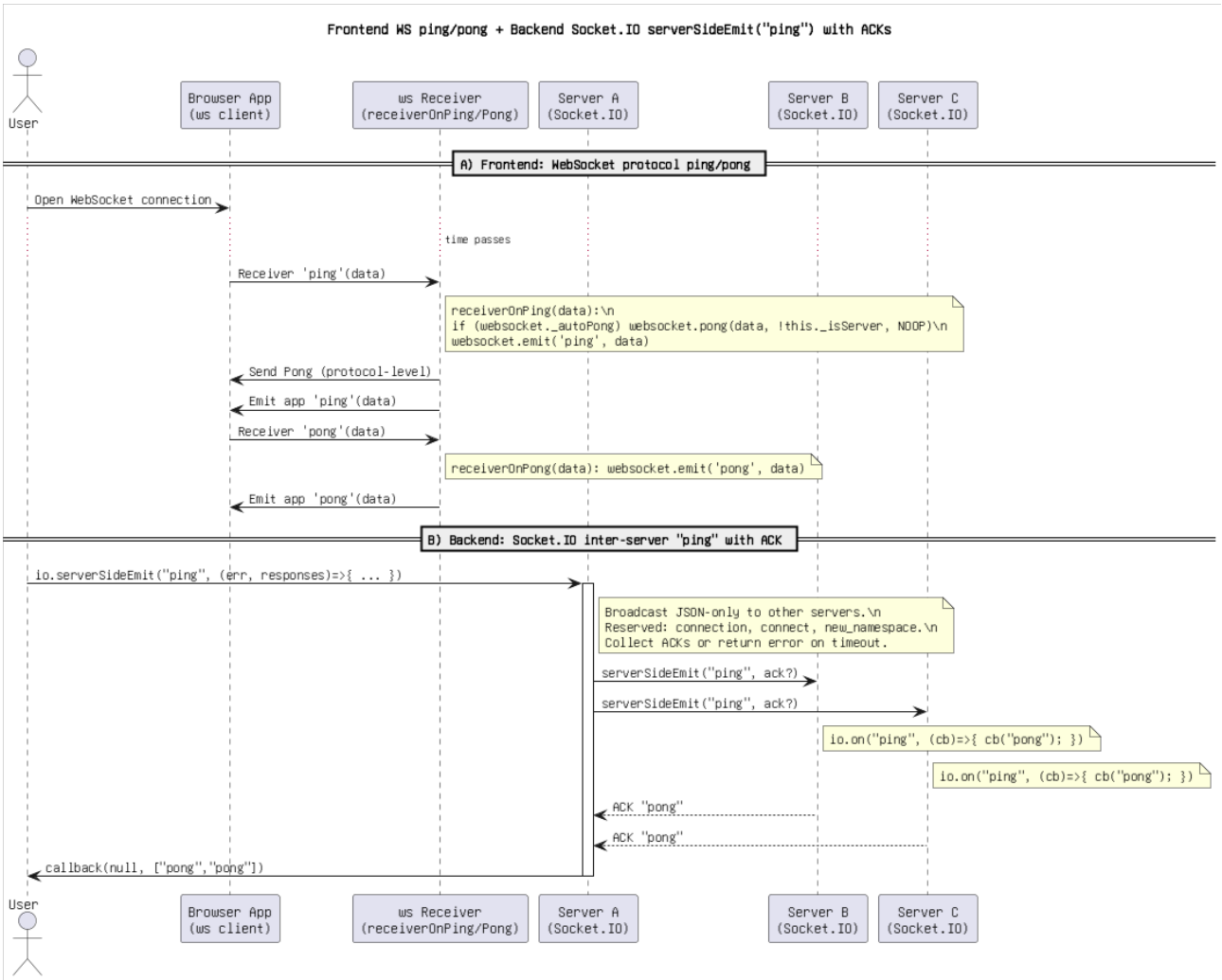


Figure 4: Frontend WS ping_pong + Backend Socket.IO serverSideEmit(_ping_) with ACKs(1).png

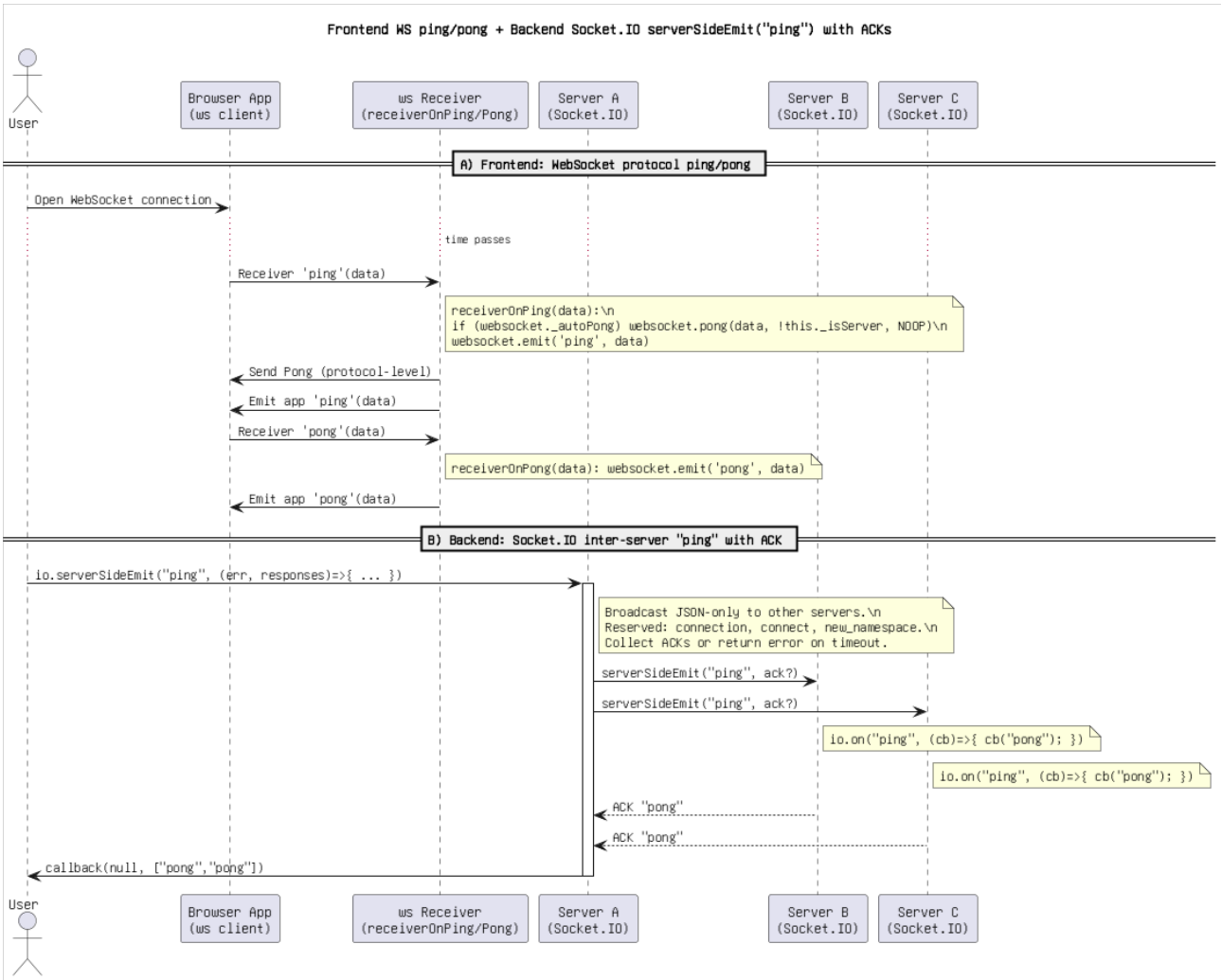


Figure 5: Frontend WS ping_pong + Backend Socket.IO serverSideEmit(_ping_) with ACKs(2).png

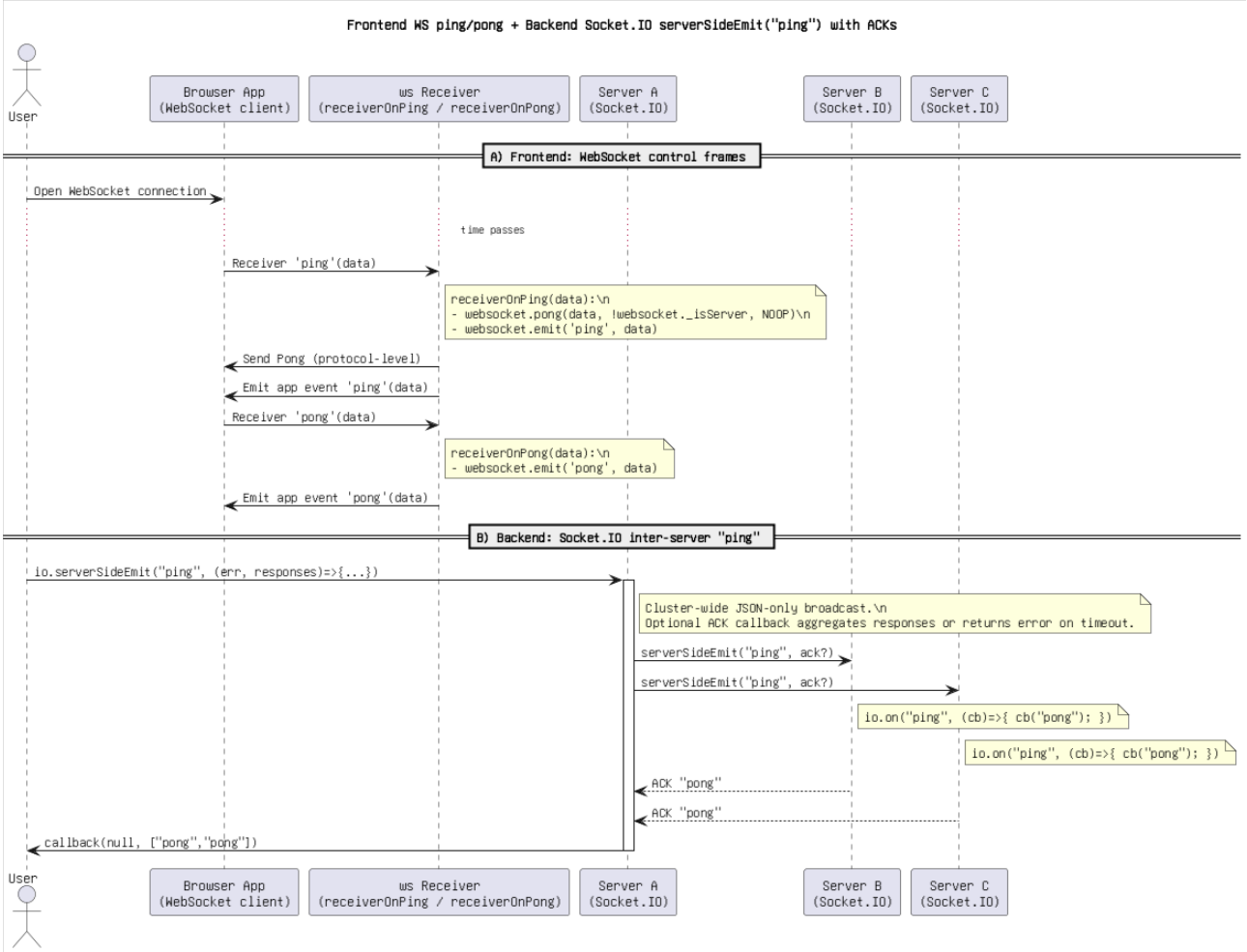


Figure 6: Frontend WS ping_pong + Backend Socket.IO serverSideEmit(_ping_) with ACKs.png

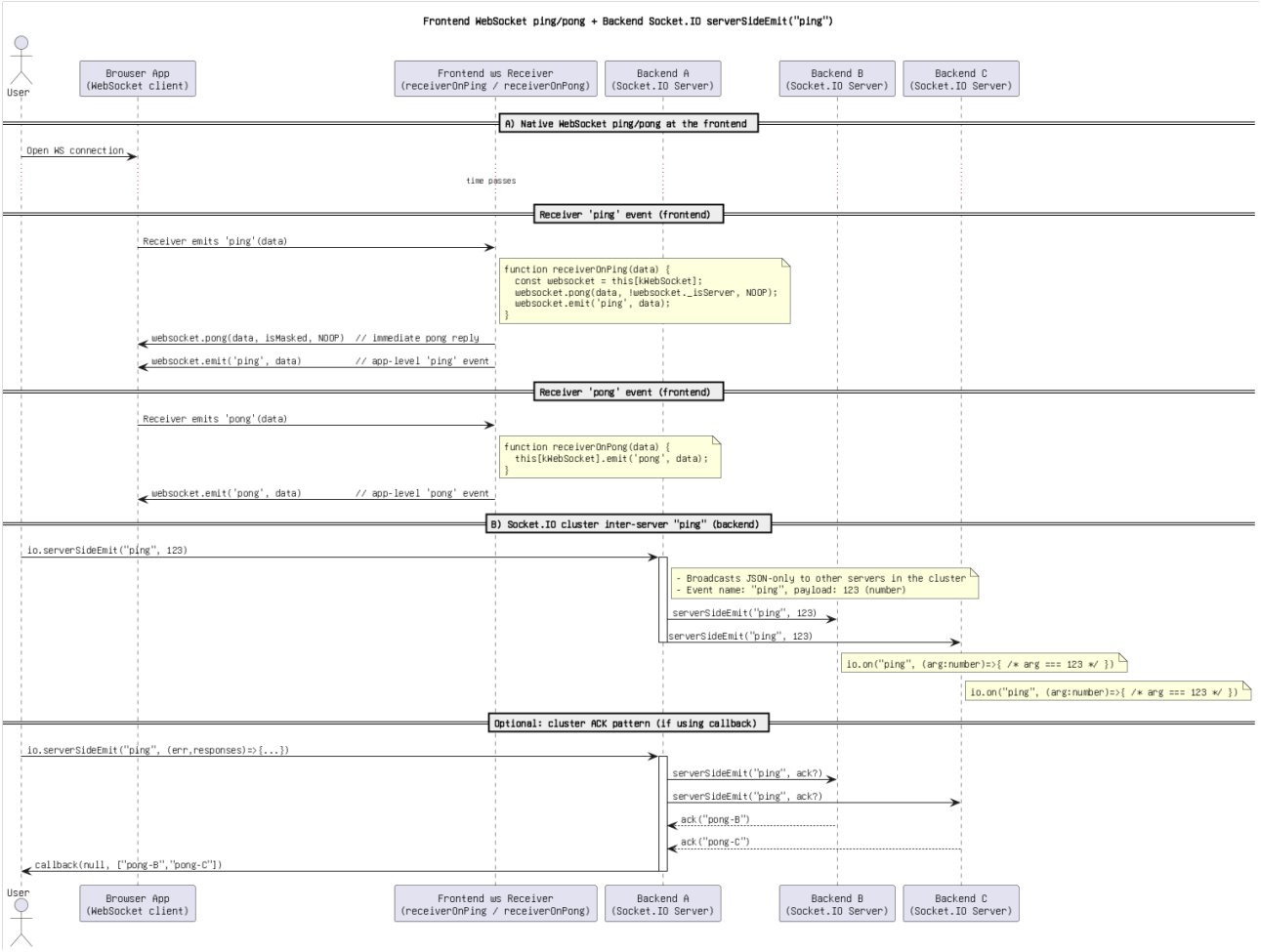


Figure 7: Frontend WebSocket ping_pong + Backend Socket.IO serverSideEmit(_ping_).png

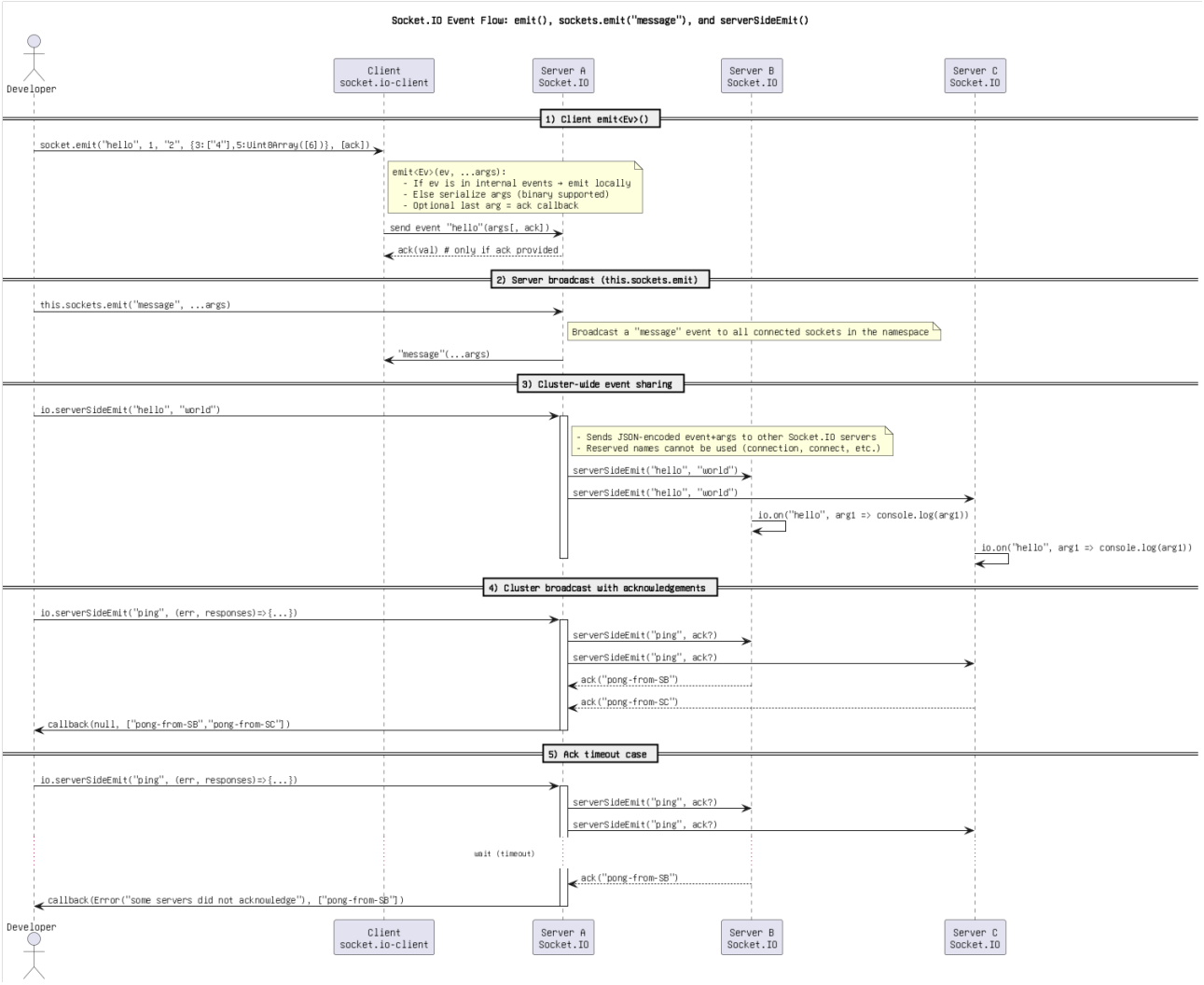


Figure 8: Socket.IO Event Flow_ emit(), sockets.emit(_message_), and serverSideEmit().png

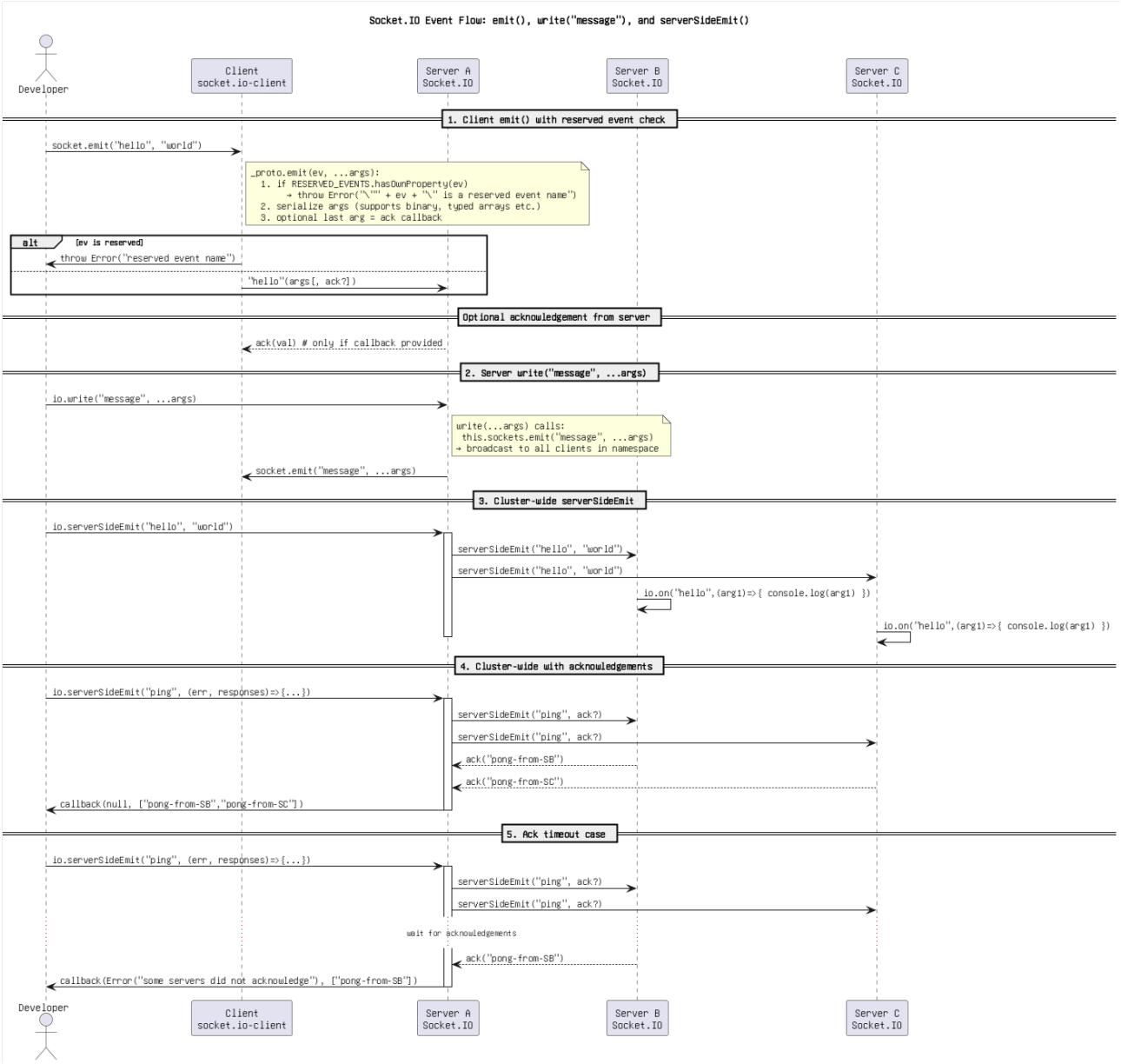


Figure 9: Socket.IO Event Flow_ emit(), write(_message_), and serverSideEmit().drawio.png

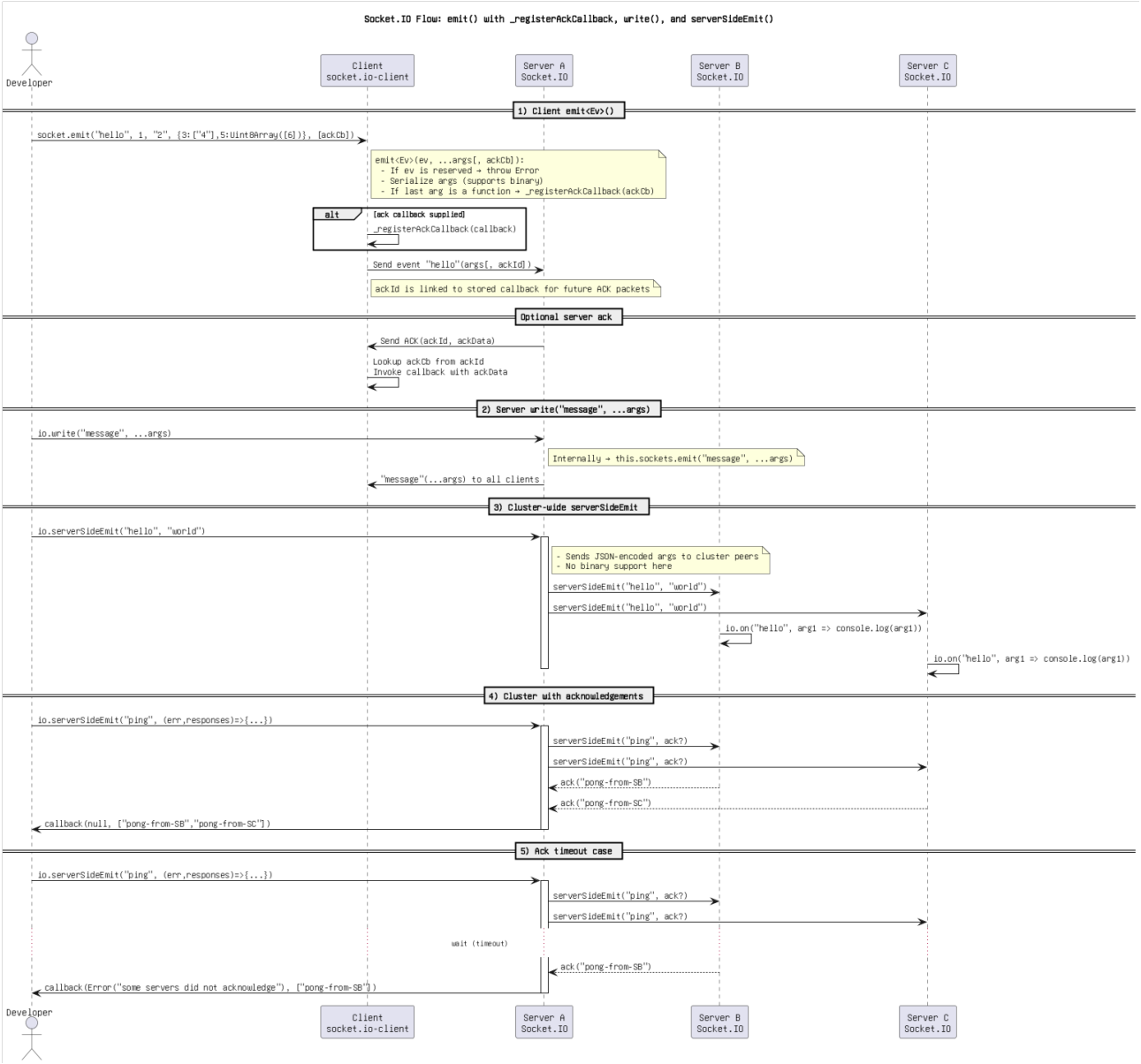


Figure 10: Socket.IO Flow_ emit() with _registerAckCallback, write(), and serverSideEmit().png

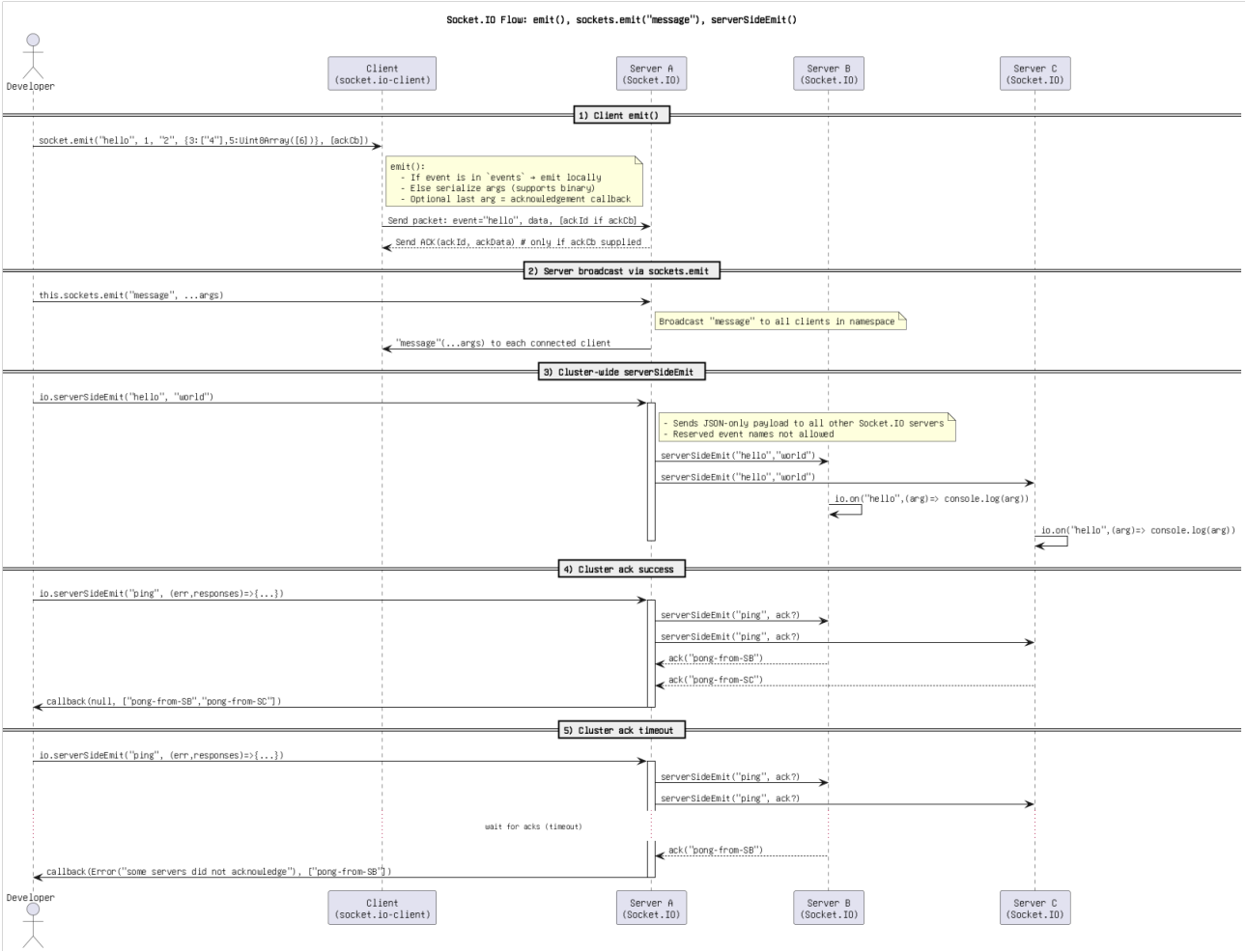


Figure 11: Socket.IO Flow_ emit(), sockets.emit(_message_), serverSideEmit().png

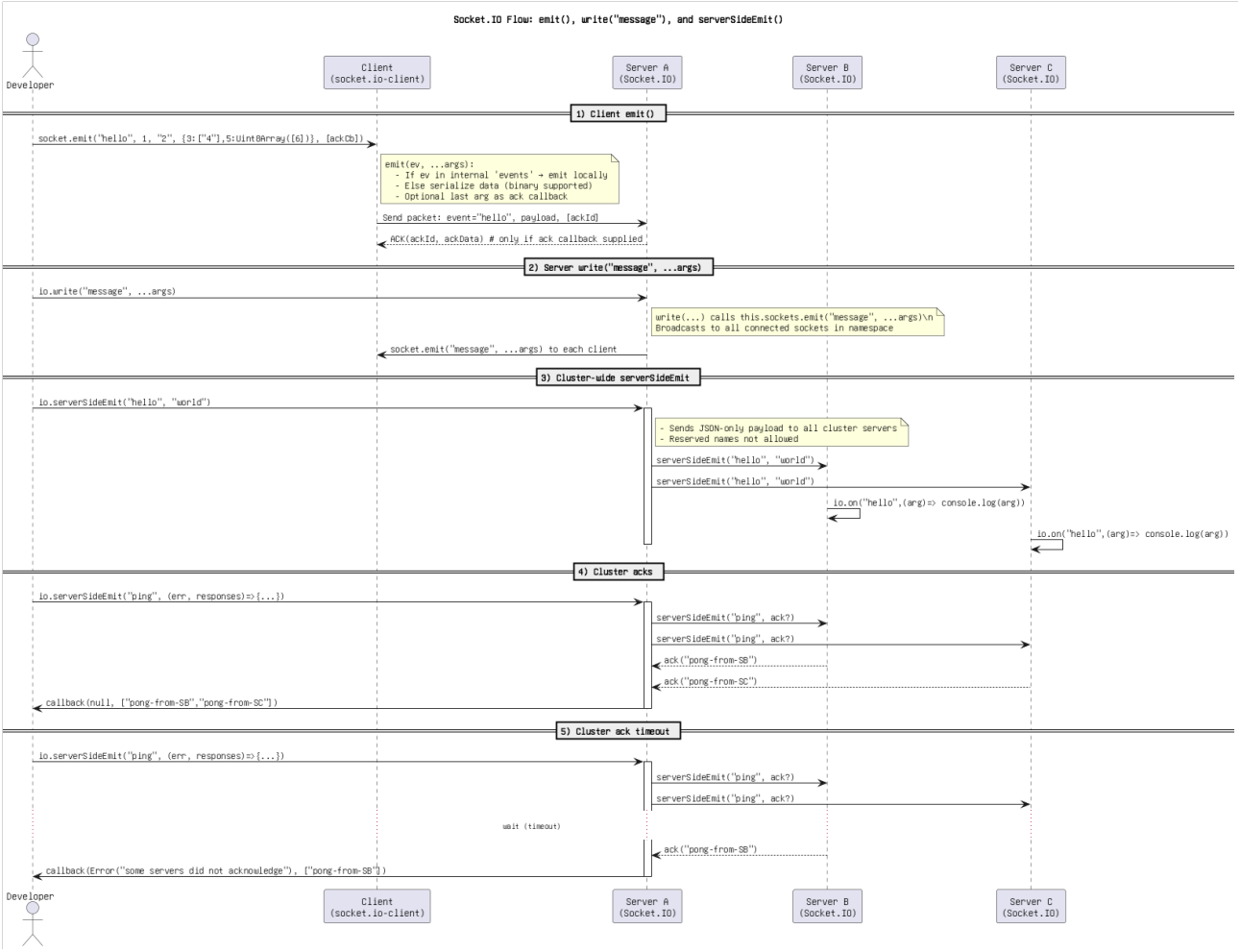


Figure 12: Socket.IO Flow_ emit(), write(_message_), and serverSideEmit().png

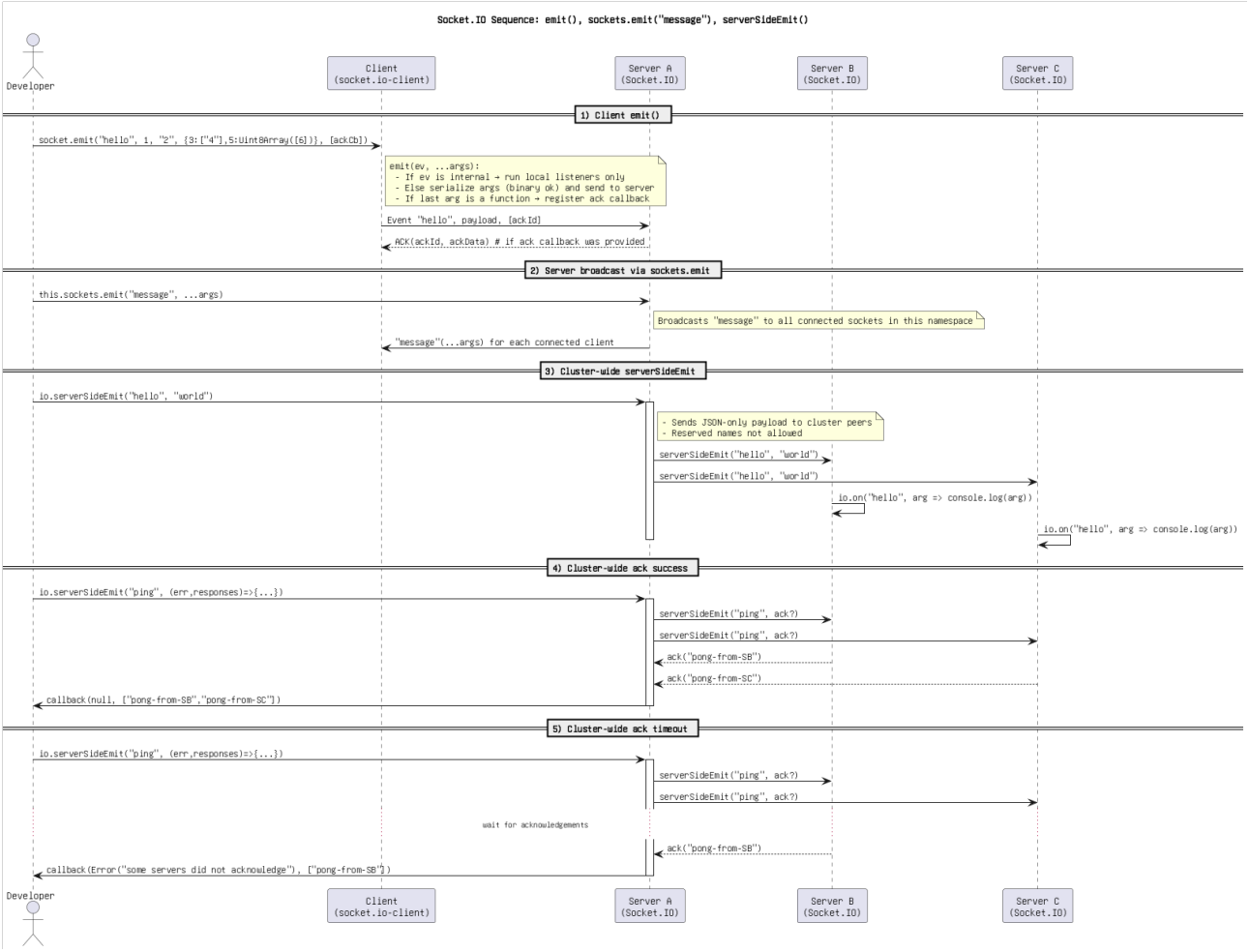


Figure 13: Socket.IO Sequence_ emit(), sockets.emit(_message_), serverSideEmit().png

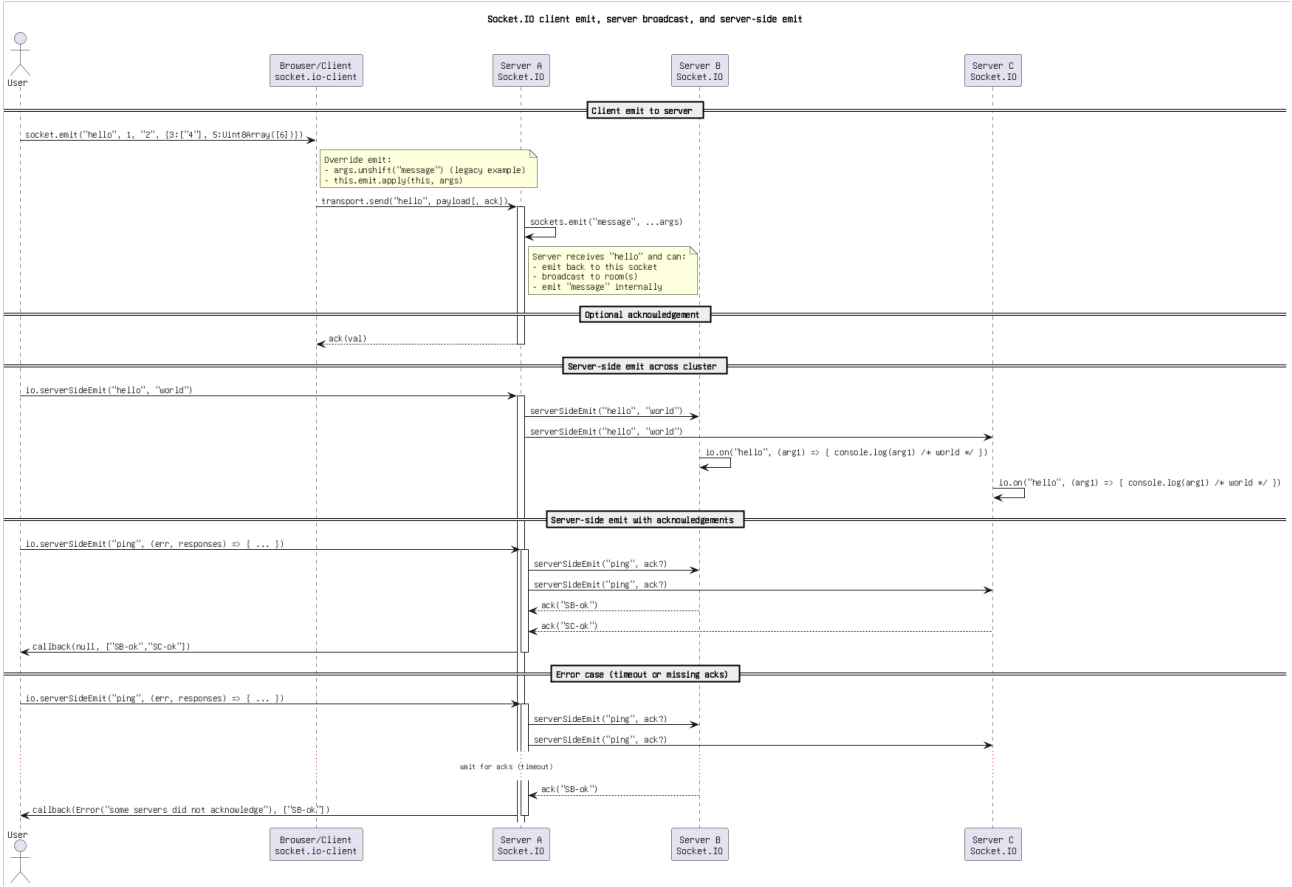
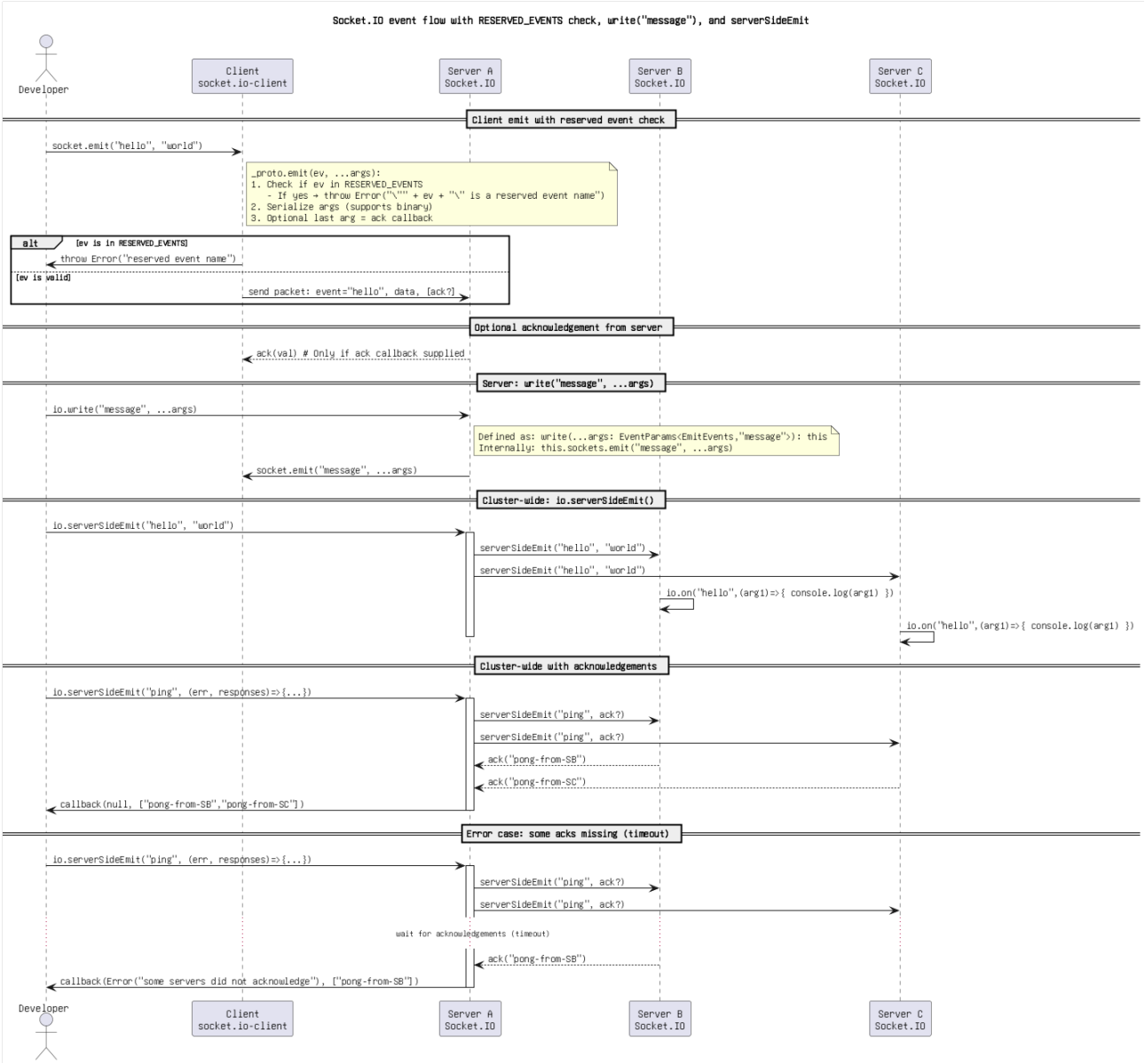
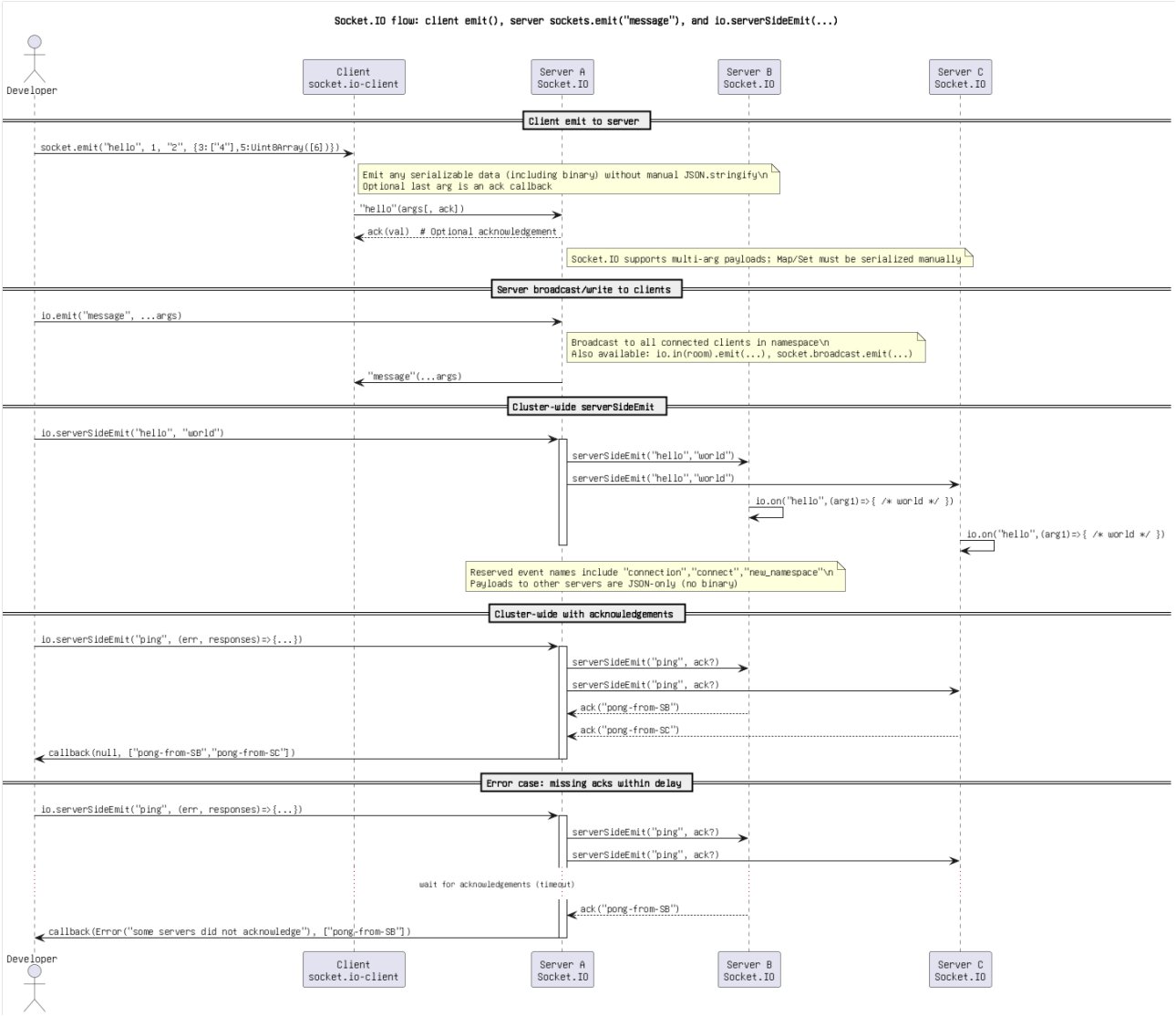


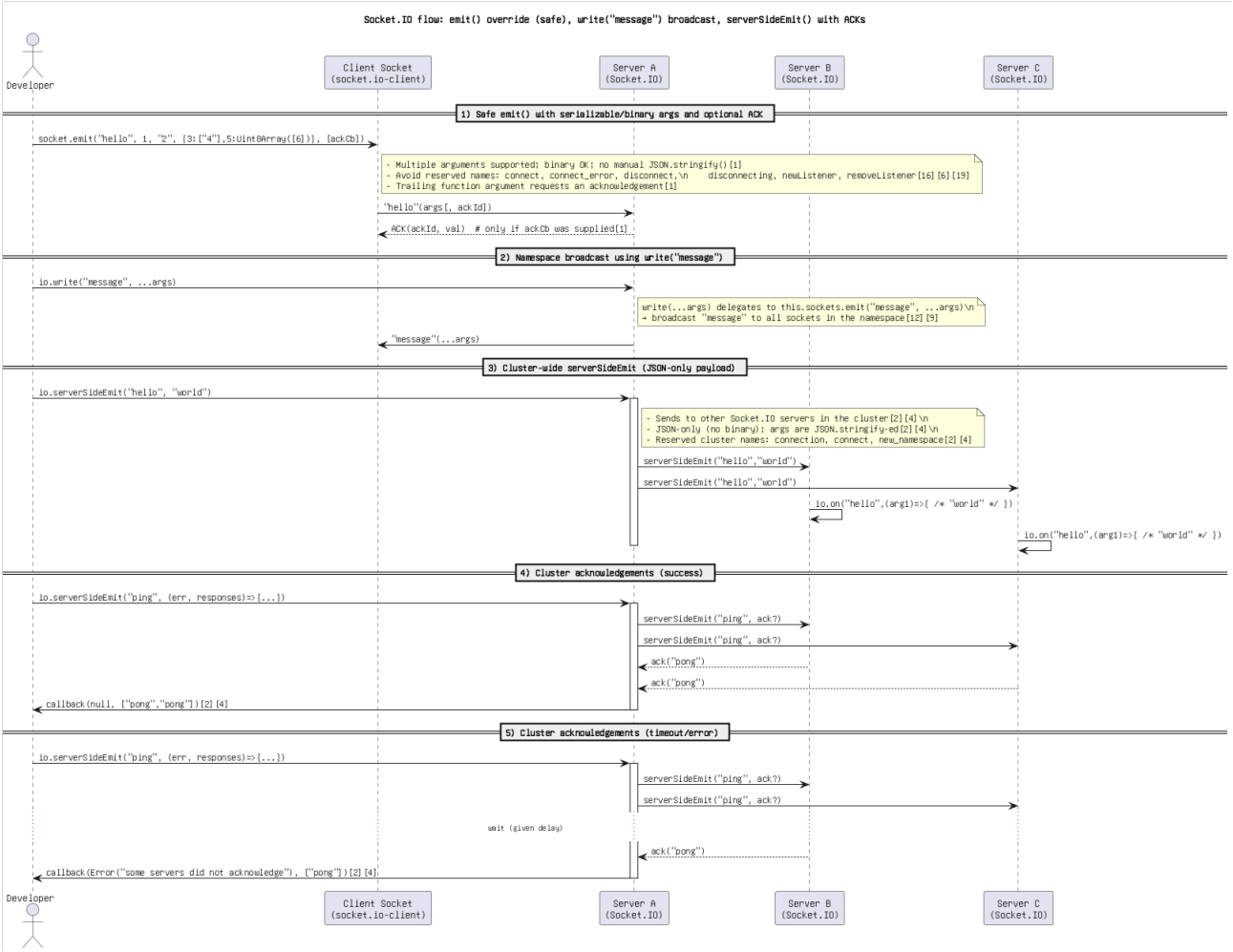
Figure 14: Socket.IO client emit, server broadcast, and server-side emit.png



15: Socket.IO event flow with RESERVED_EVENTS check, write(_message_), and serverSideEmit.dra



e 16: Socket.IO flow_ client emit(), server sockets.emit(_message_), and io.serverSideEmit(...).drawi



e 17: Socket.IO flow_ emit() override (safe), write(_message_) broadcast, serverSideEmit() with ACK

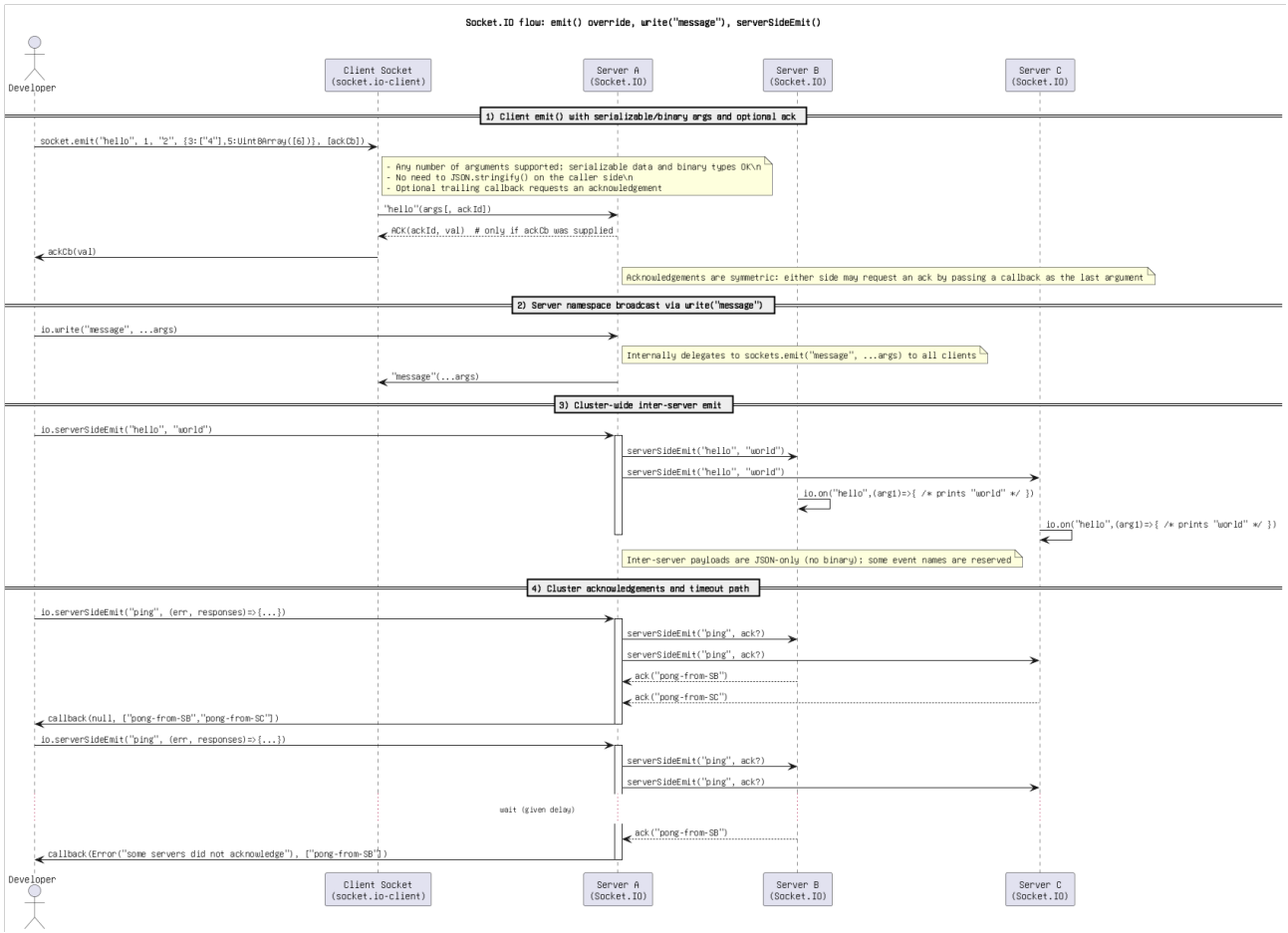


Figure 18: Socket.IO flow_ emit() override, write(_message_), serverSideEmit().png

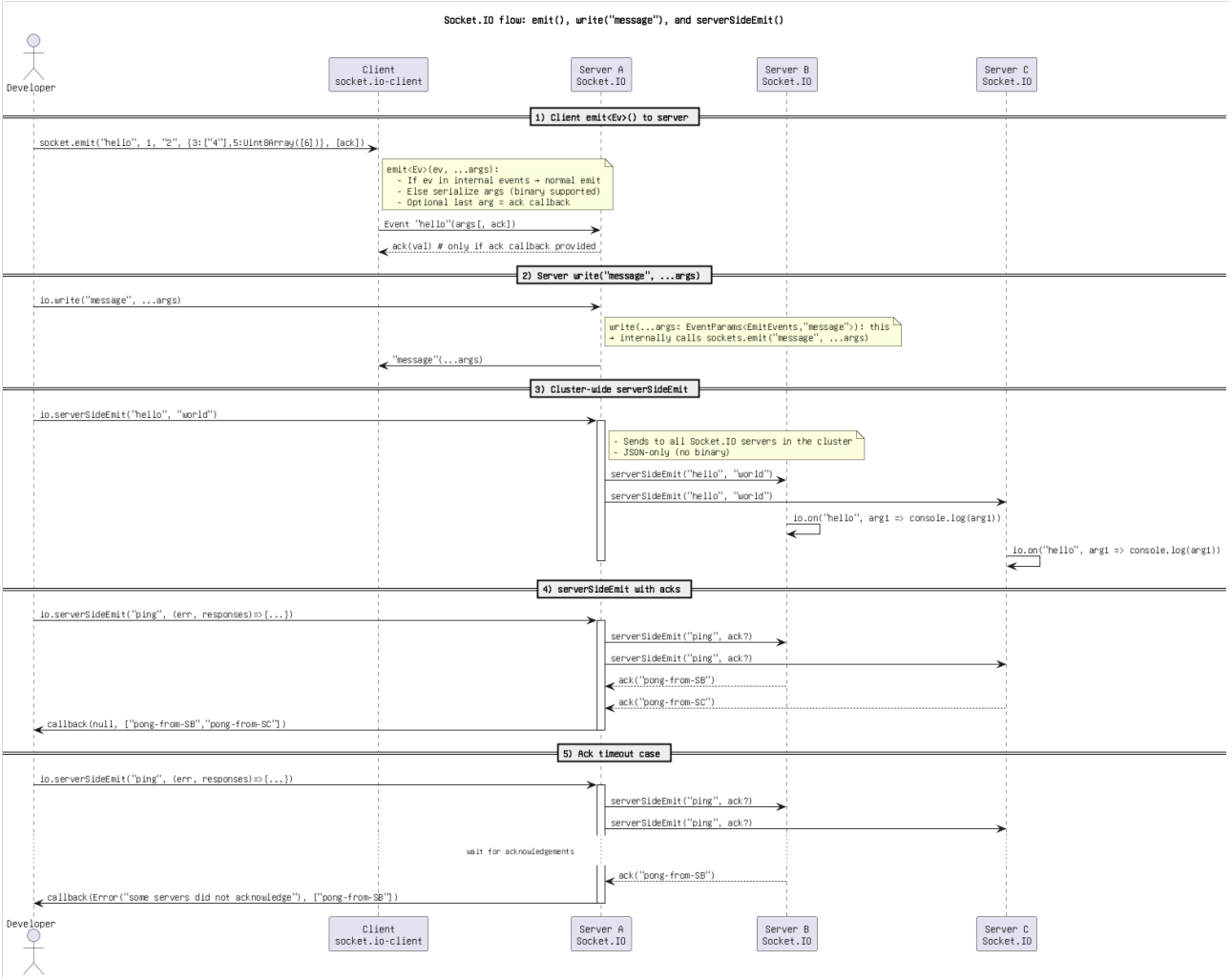


Figure 19: Socket.IO flow_ emit(), write(_message_), and serverSideEmit().png

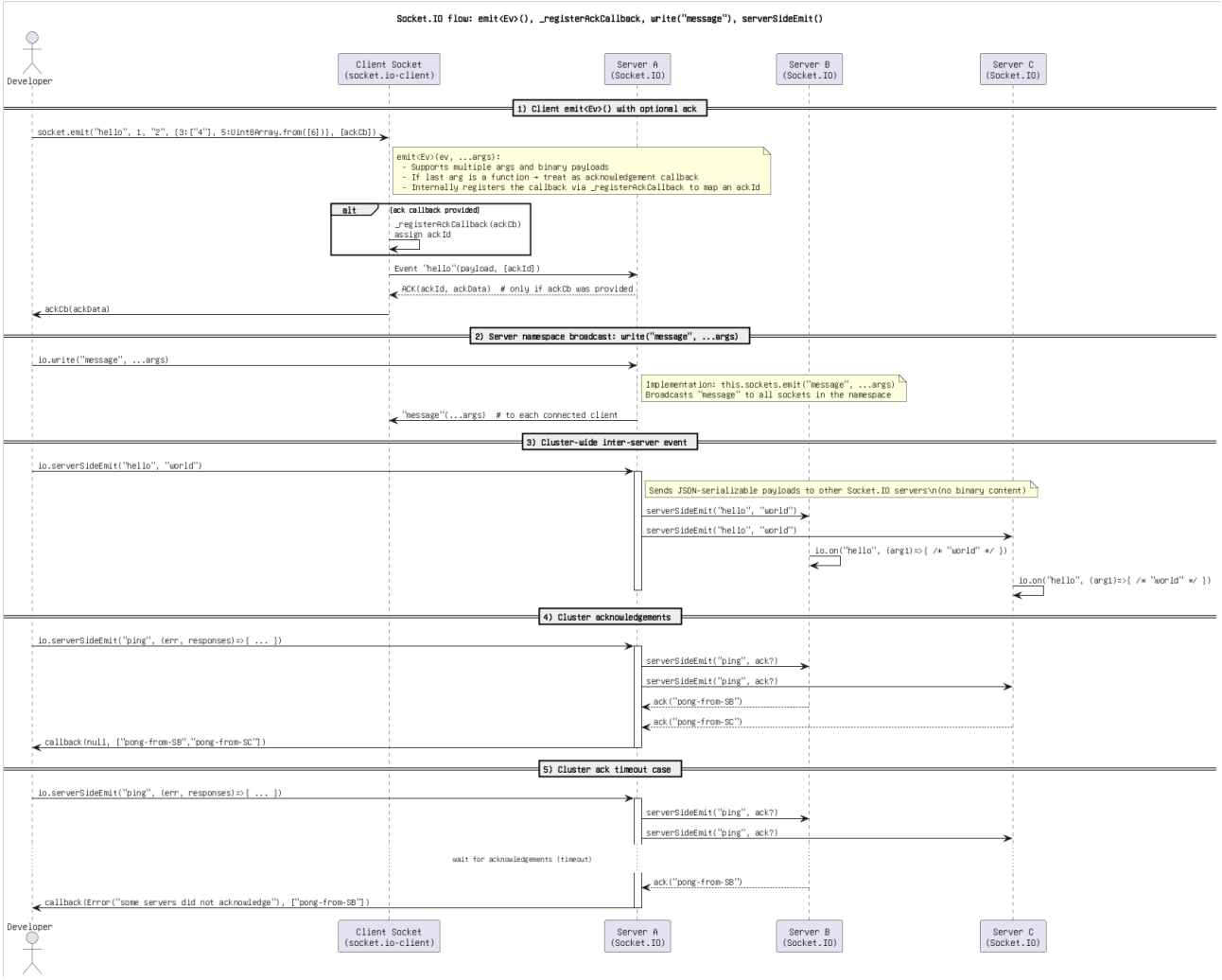
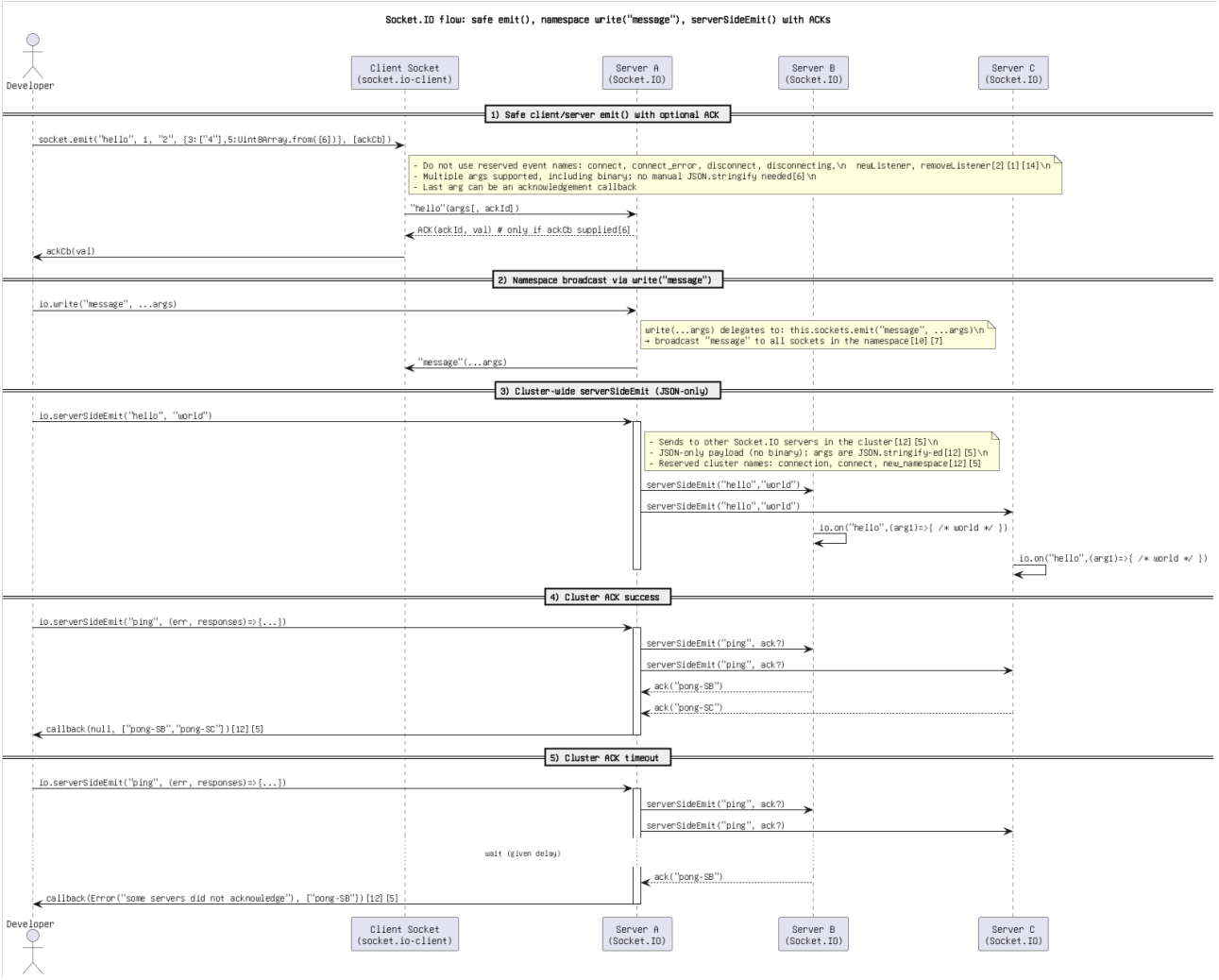
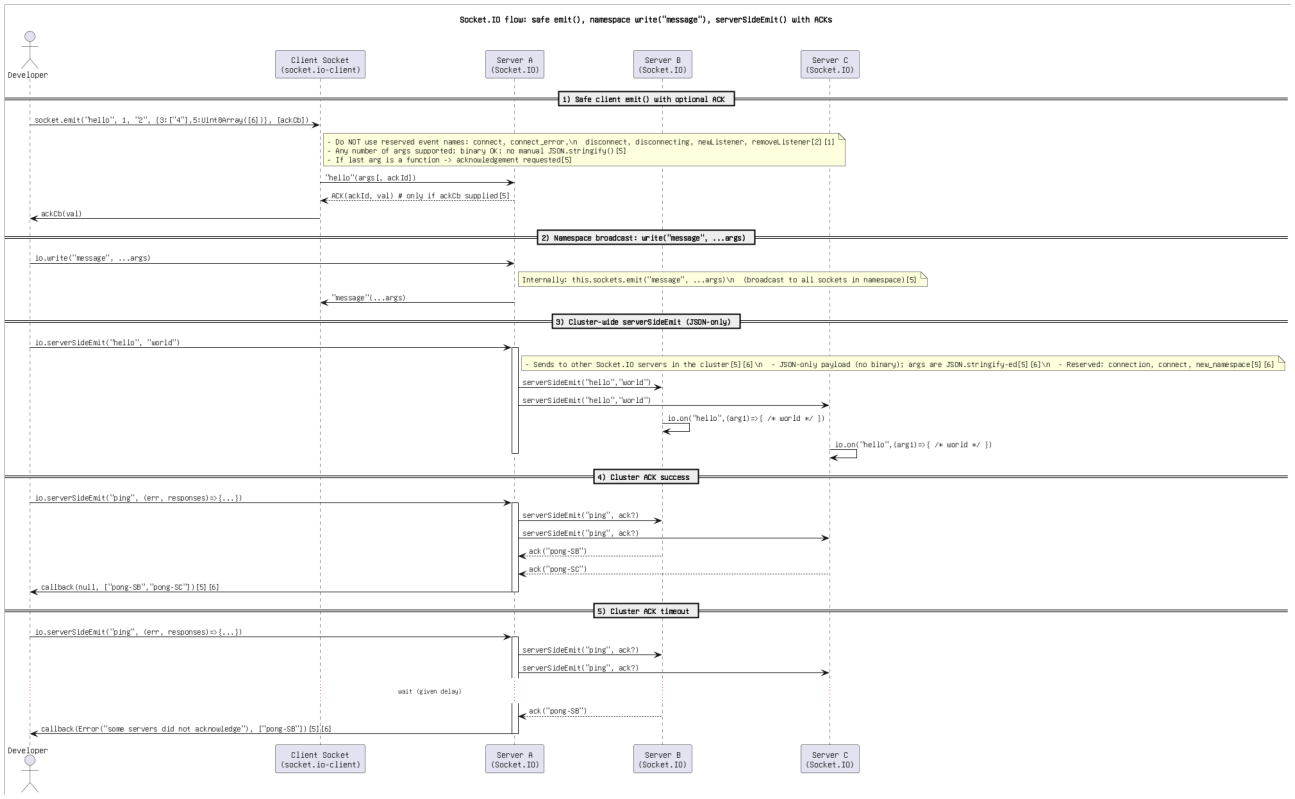


Figure 20: Socket.IO flow_ emit_Ev_(), _registerAckCallback, write(_message_), serverSideEmit(.png



ure 21: Socket.IO flow_ safe emit(), namespace write(_message_), serverSideEmit() with ACKs(1).p



igure 22: Socket.IO flow_ safe emit(), namespace write(_message_), serverSideEmit() with ACKs.pn

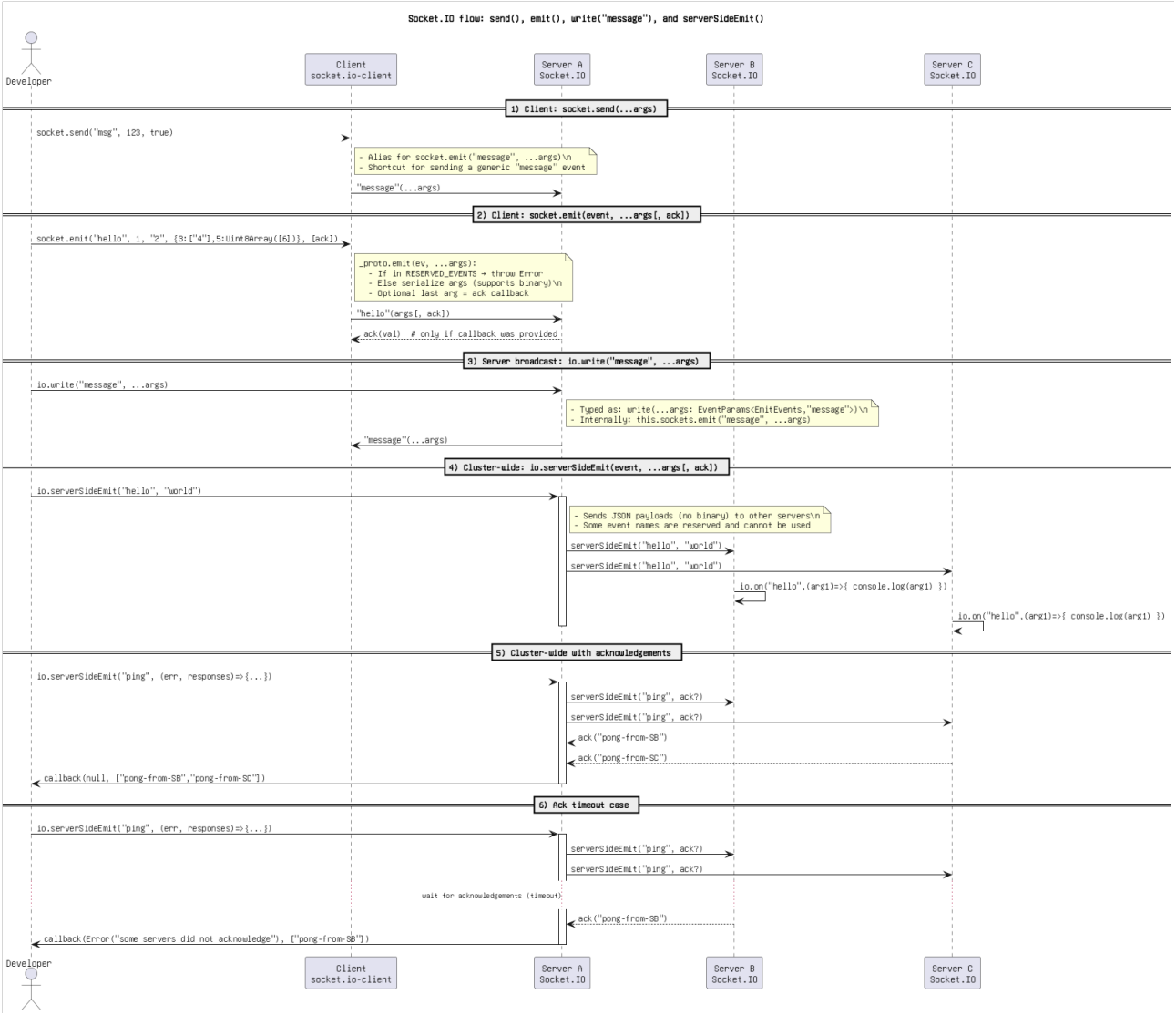


Figure 23: Socket.IO flow_ send(), emit(), write(_message_), and serverSideEmit().png

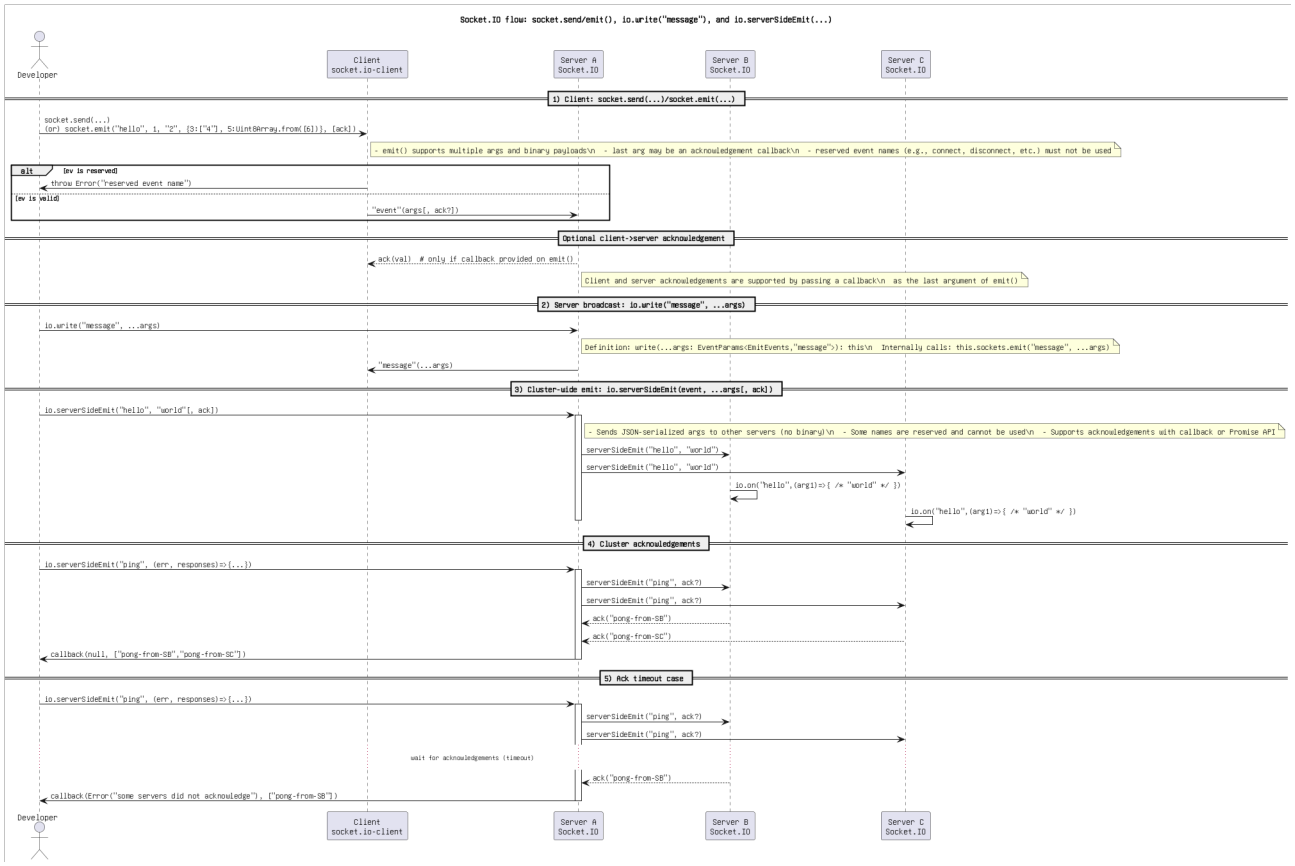


Figure 24: Socket.IO flow_ socket.send_emit(), io.write(_message_), and io.serverSideEmit(...).png

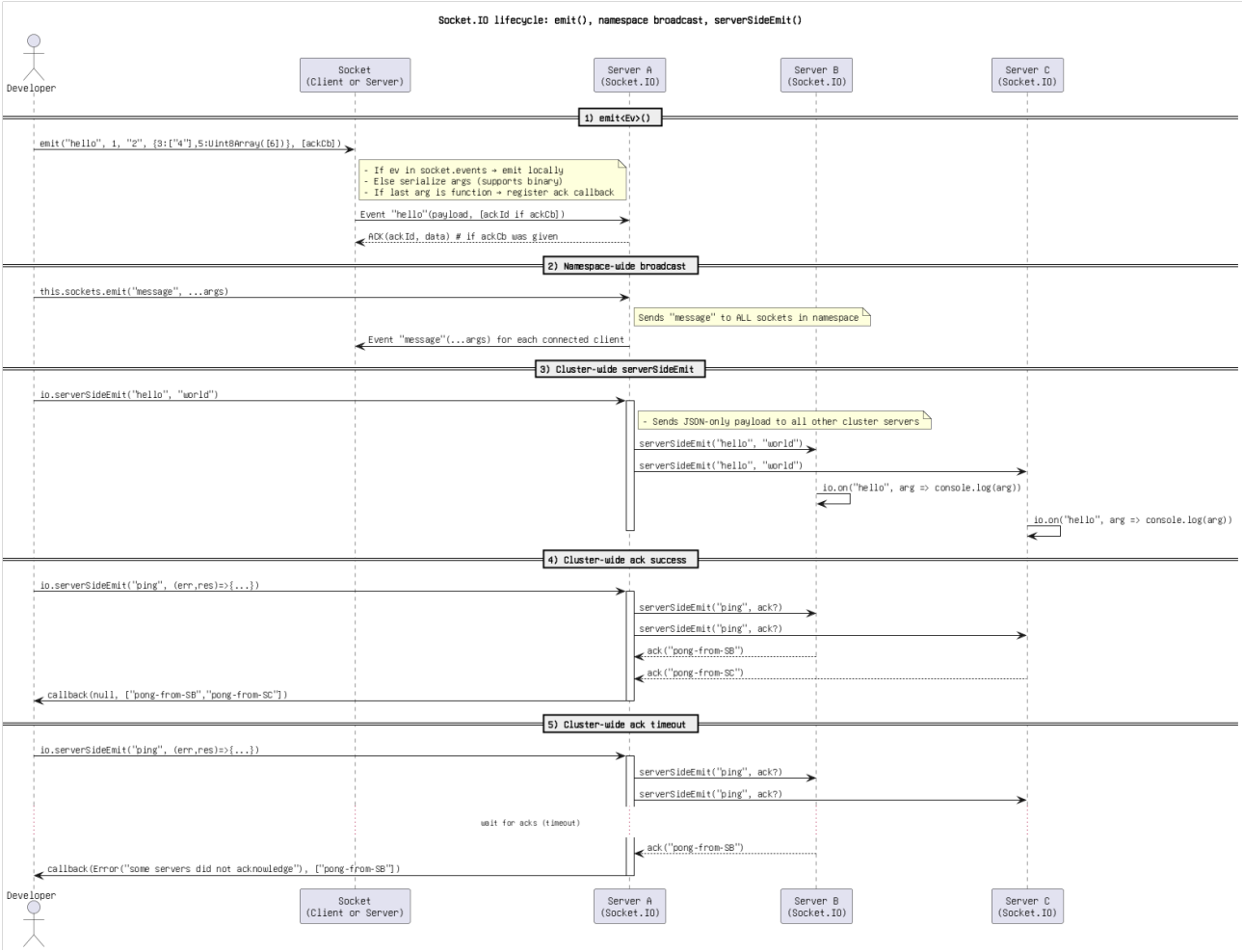


Figure 25: Socket.IO lifecycle_ emit(), namespace broadcast, serverSideEmit().png

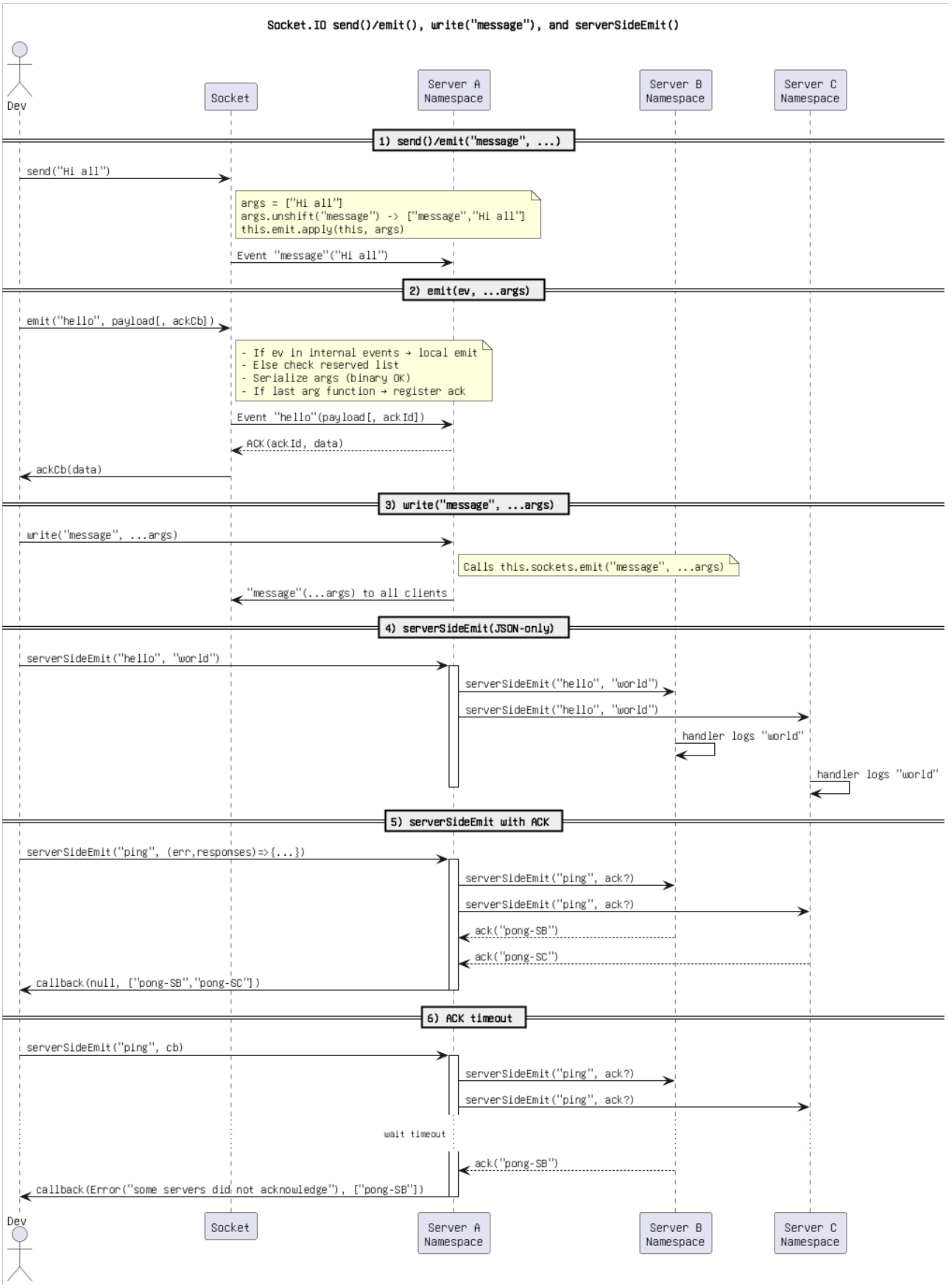


Figure 26: Socket.IO send()_emit(), write(_message_), and serverSideEmit().png

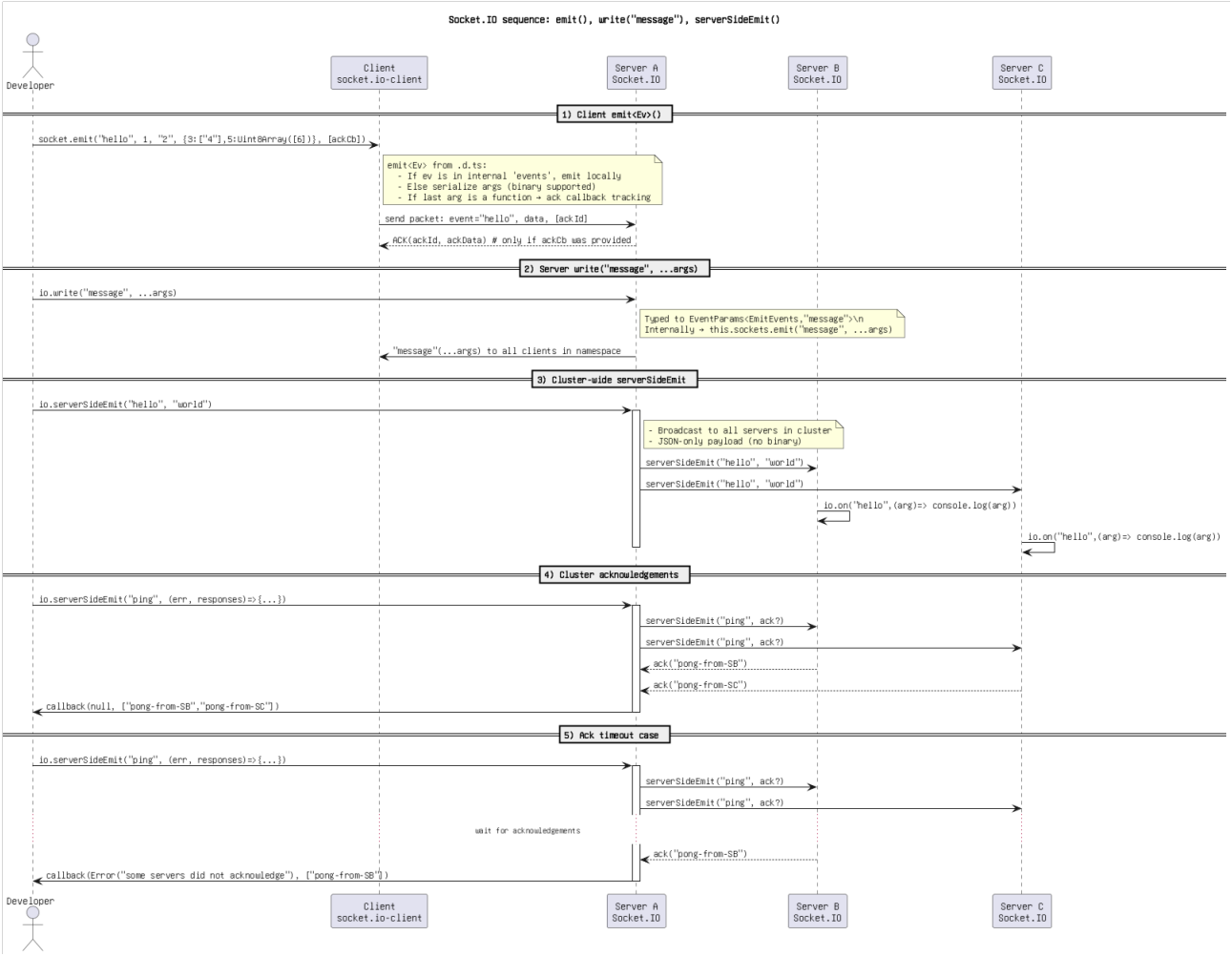


Figure 27: Socket.IO sequence_ emit(), write(_message_), serverSideEmit().png

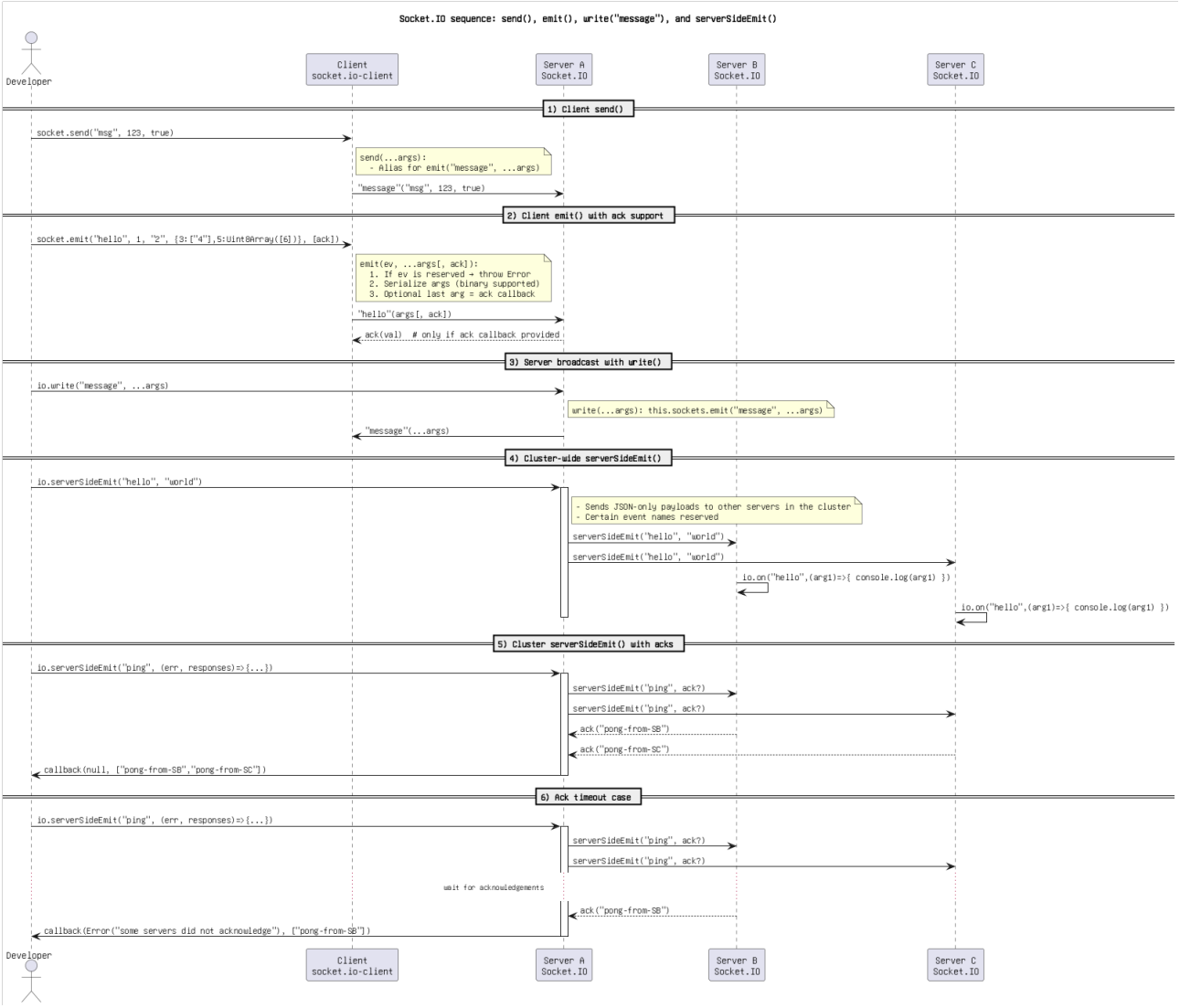


Figure 28: Socket.IO sequence_ send(), emit(), write(_message_), and serverSideEmit().png

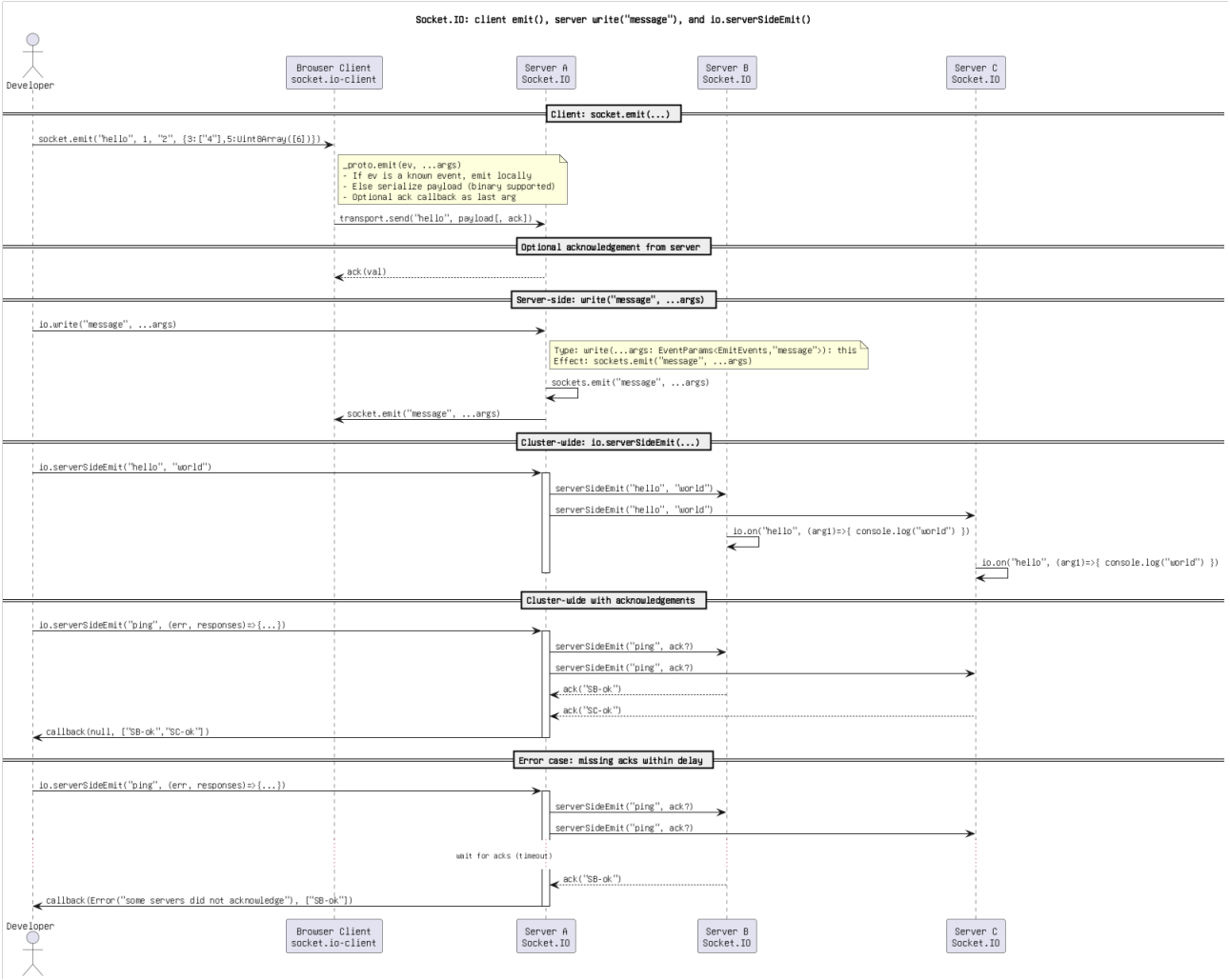


Figure 29: Socket.IO_ client emit(), server write(_message_), and io.serverSideEmit().png

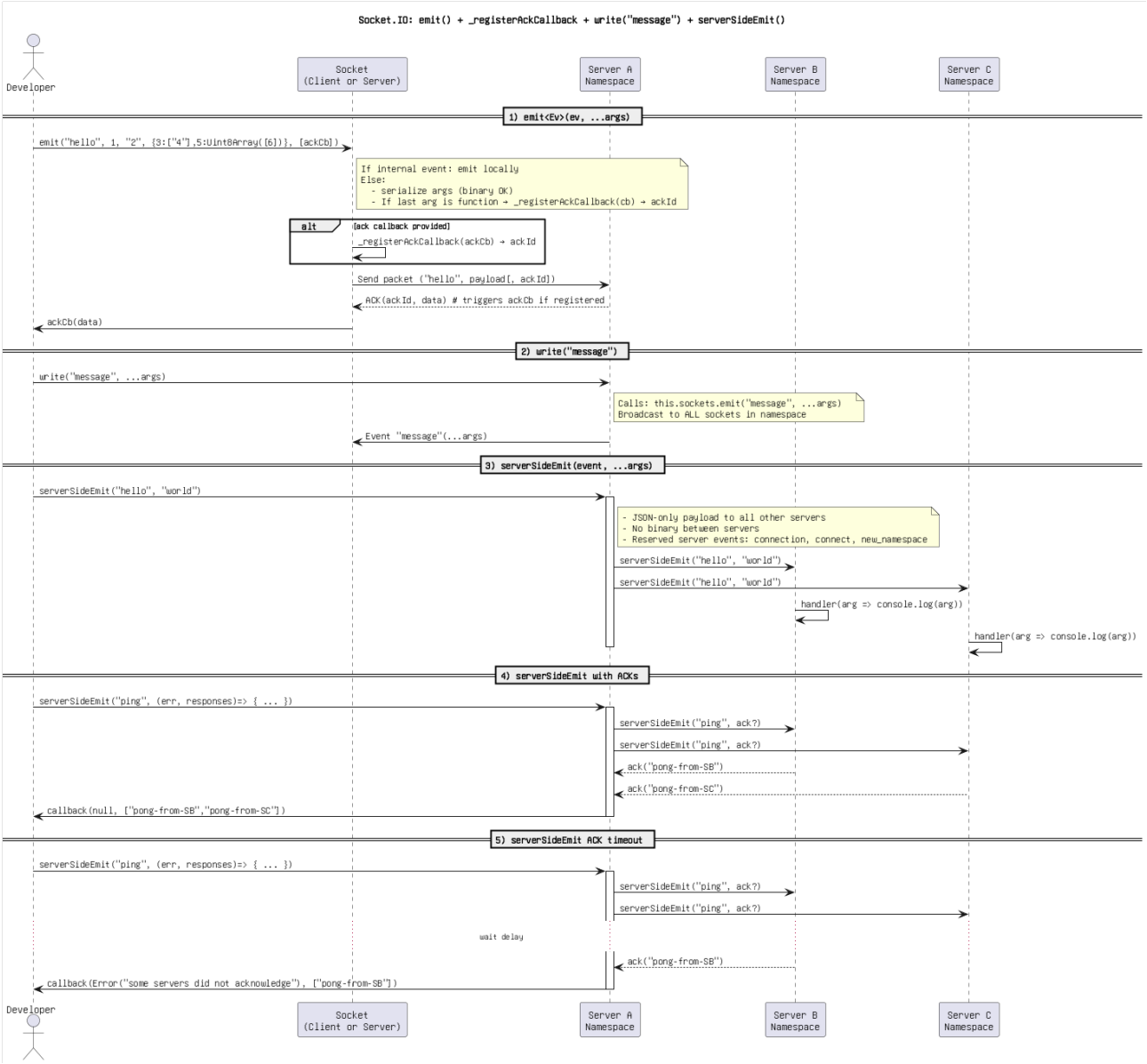
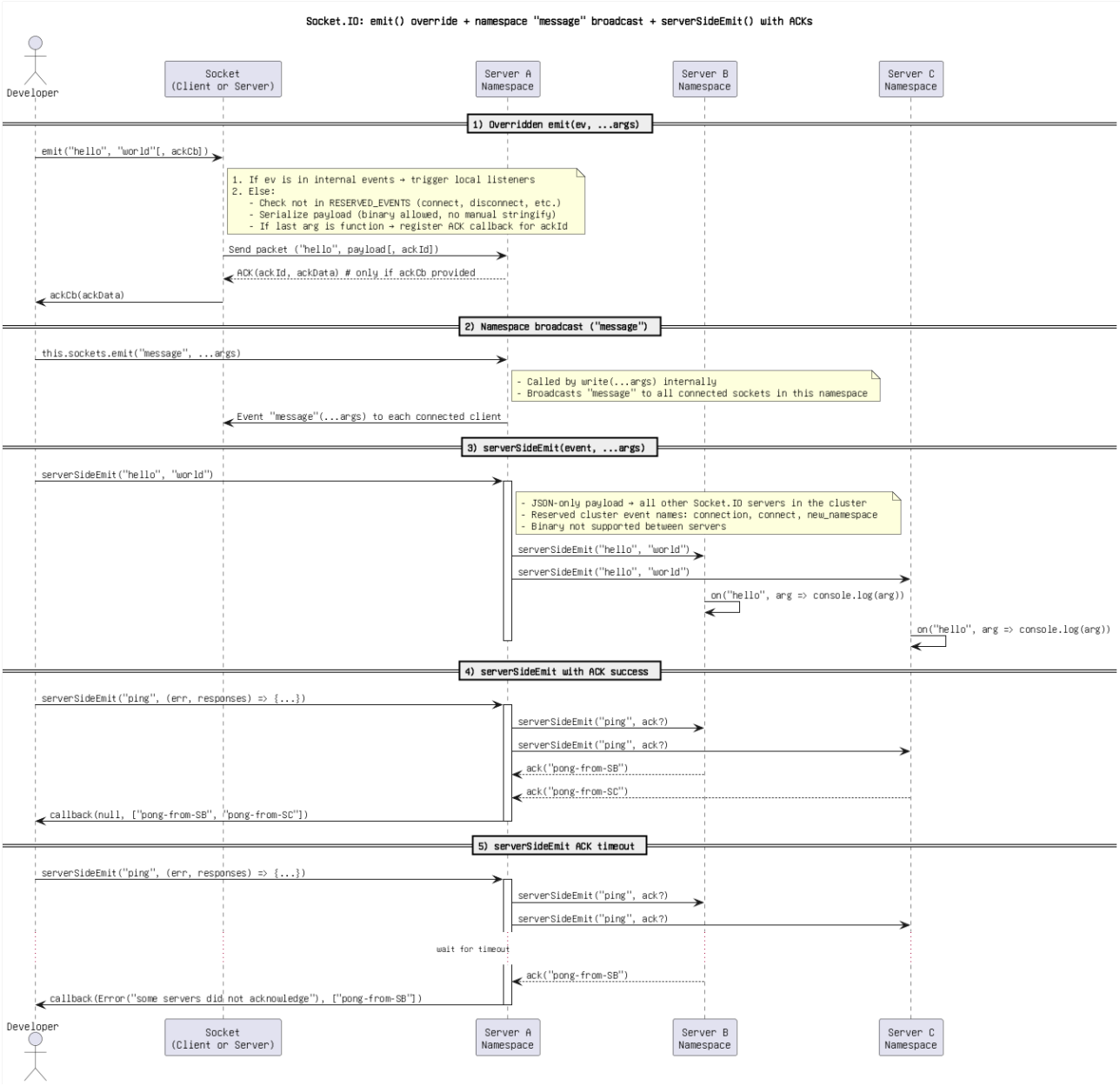


Figure 30: Socket.IO_ emit() + _registerAckCallback + write(_message_) + serverSideEmit().png



re 31: Socket.IO_ emit() override + namespace _message_ broadcast + serverSideEmit() with ACKs

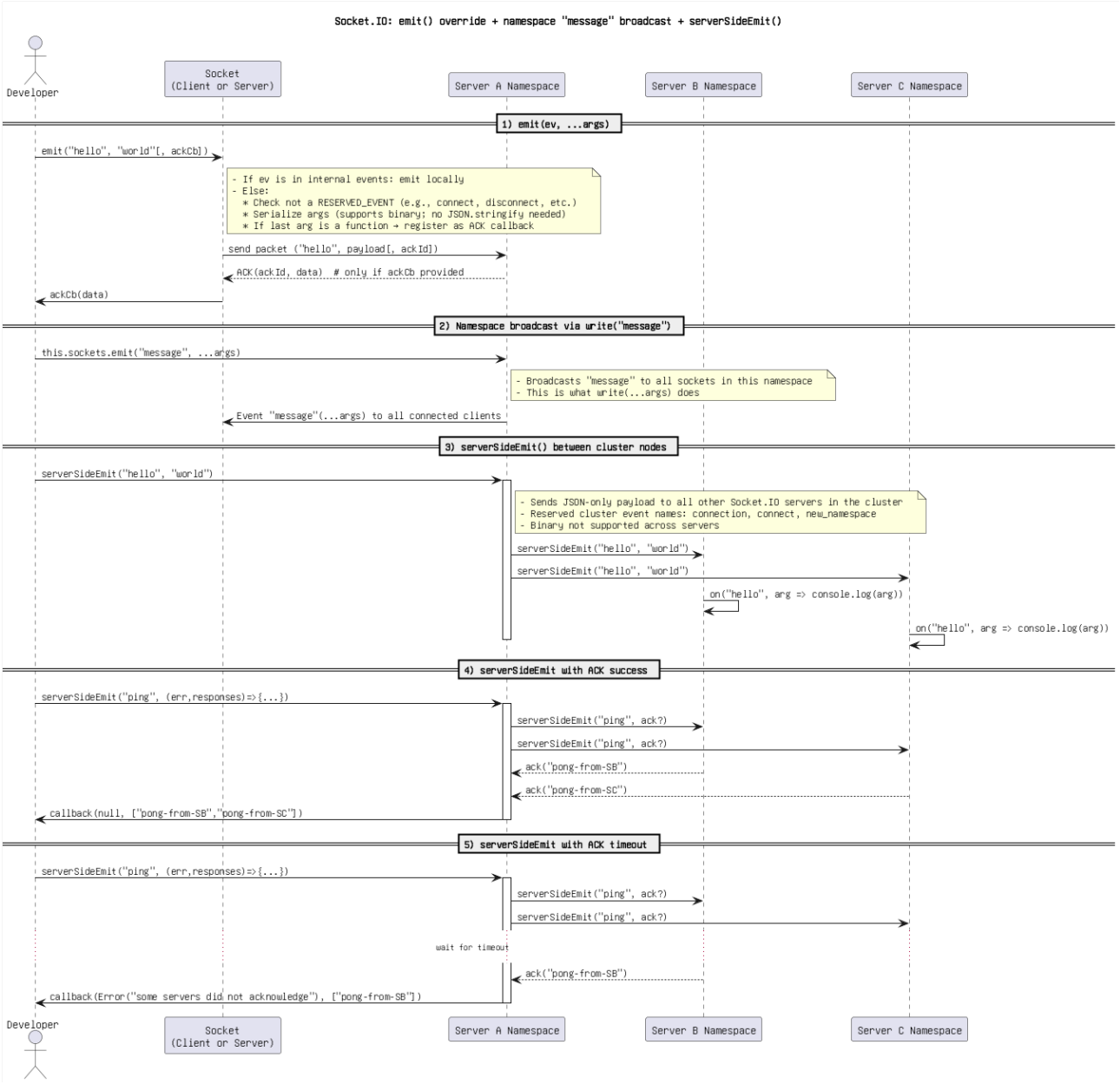
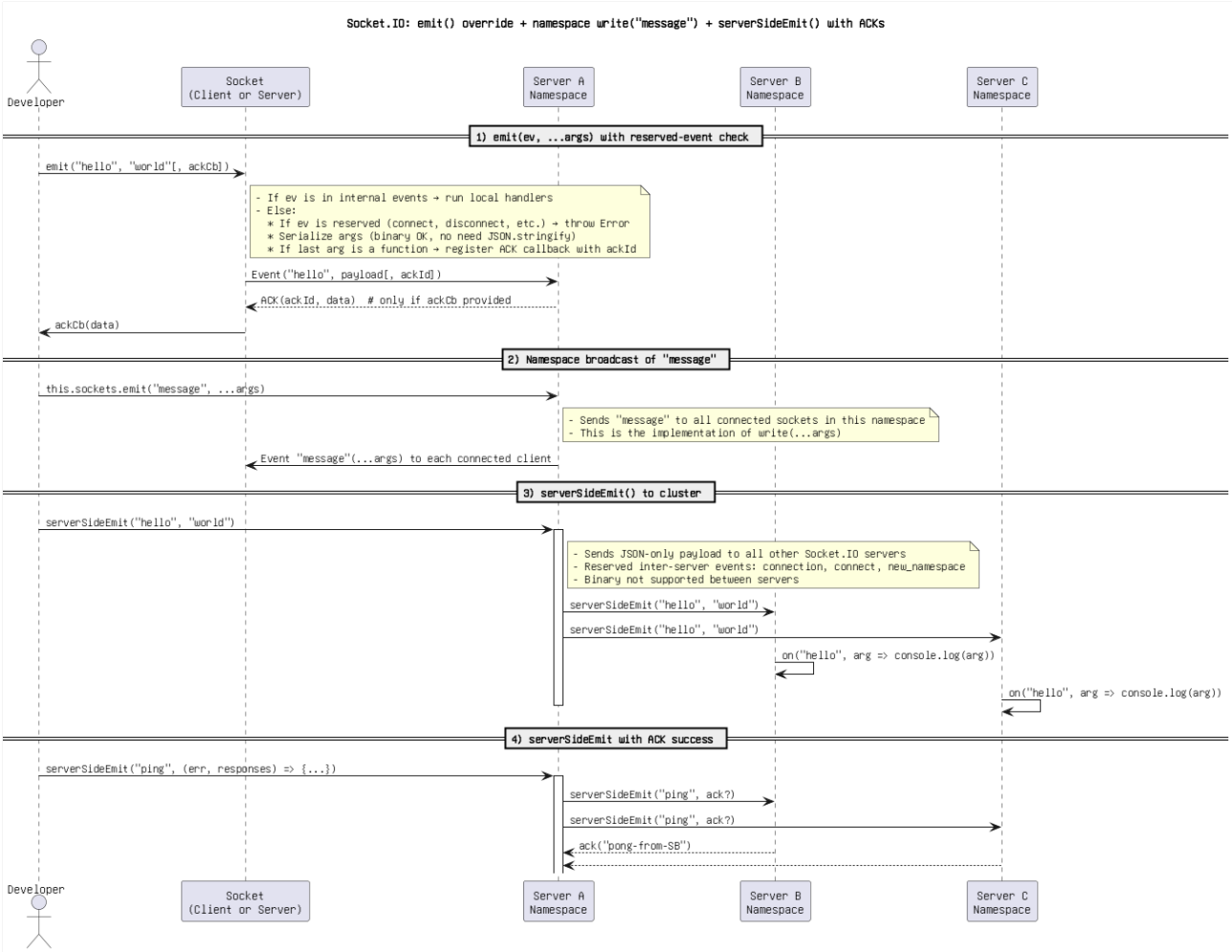


Figure 32: Socket.IO_ emit() override + namespace _message_ broadcast + serverSideEmit().png



gure 33: Socket.IO_ emit() override + namespace write(_message_) + serverSideEmit() with ACKs.p

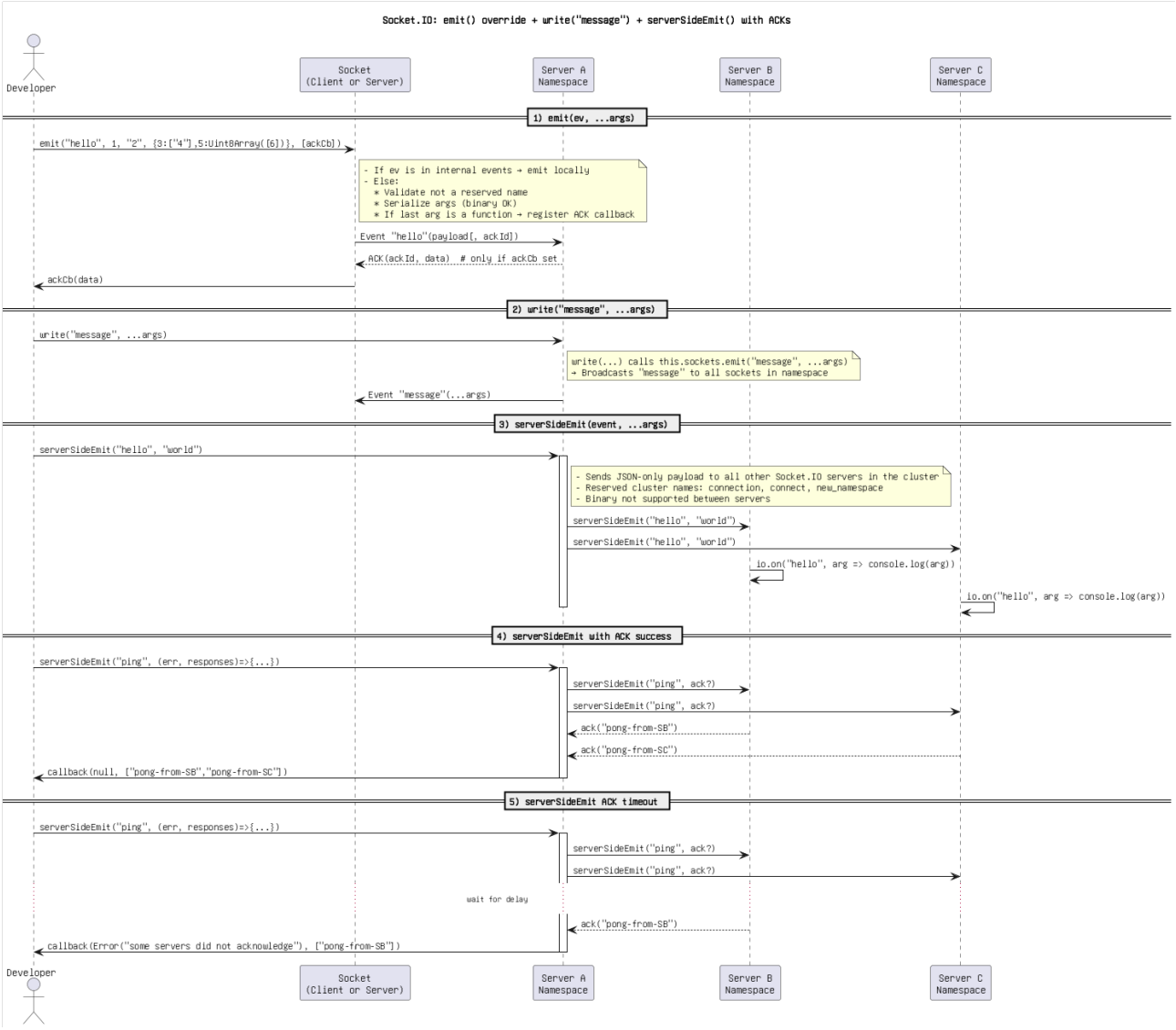


Figure 34: Socket.IO_ emit() override + write(_message_) + serverSideEmit() with ACKs(1).png

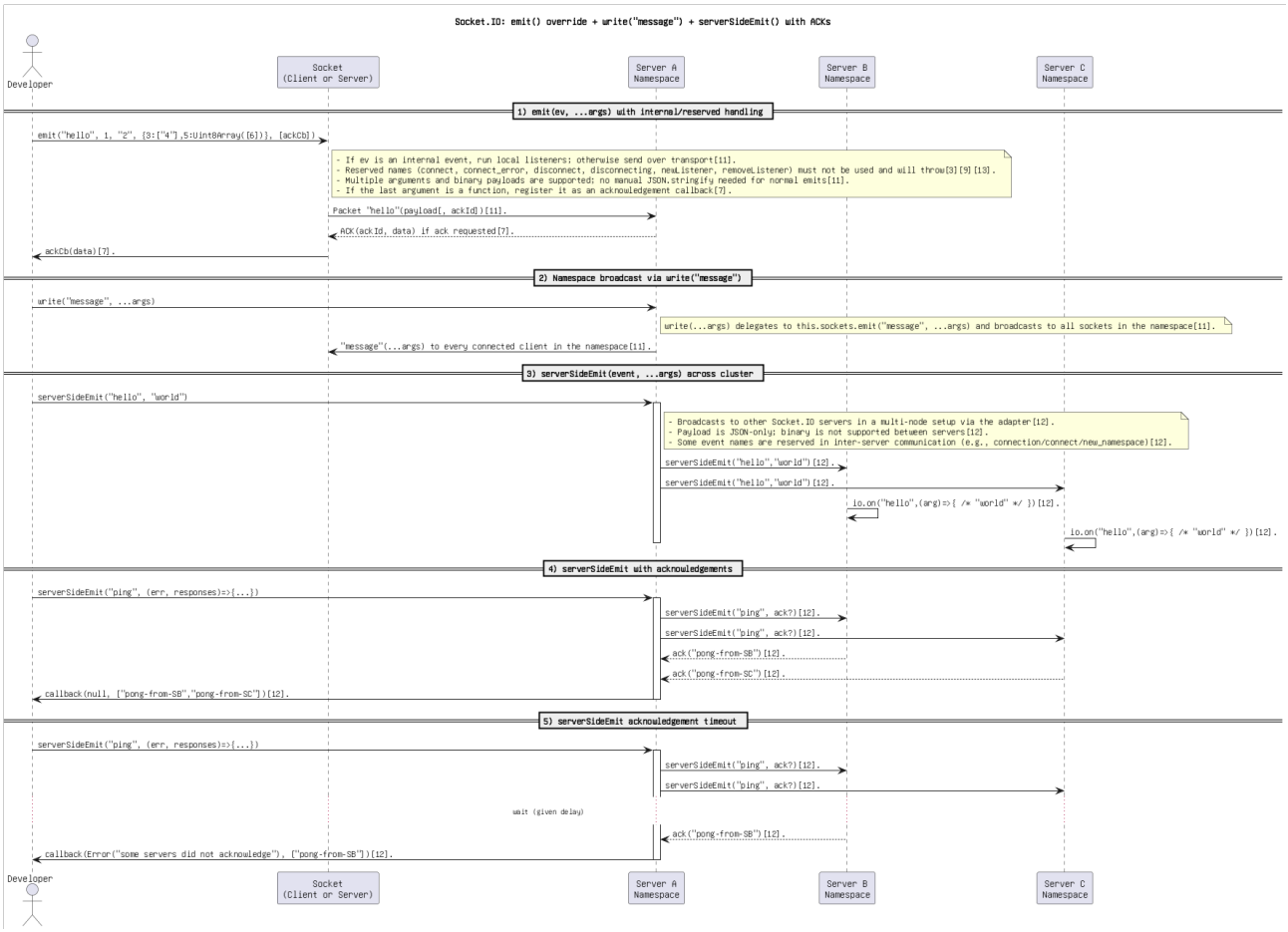


Figure 35: Socket.IO_ emit() override + write(_message_) + serverSideEmit() with ACKs(2).png

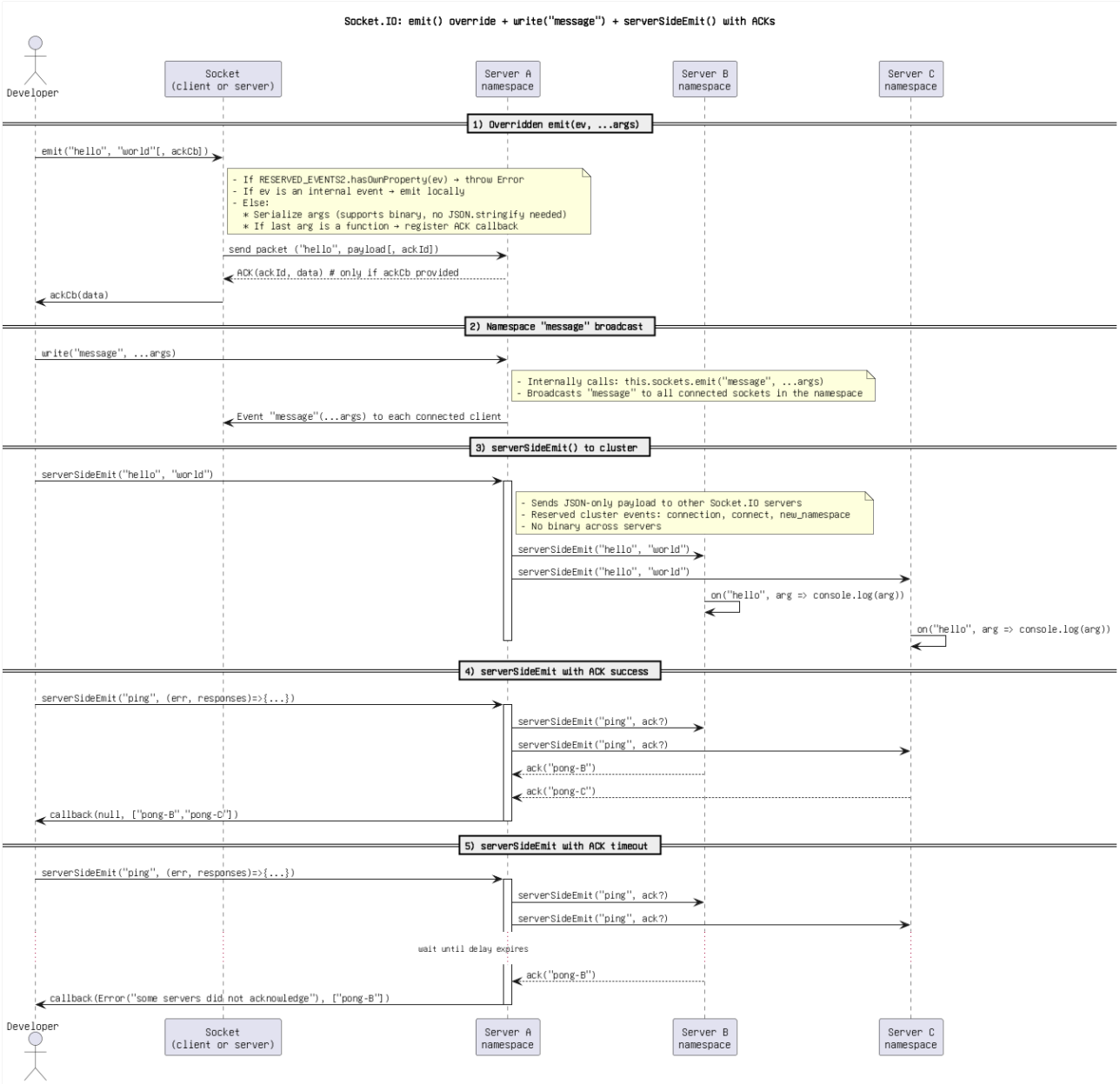
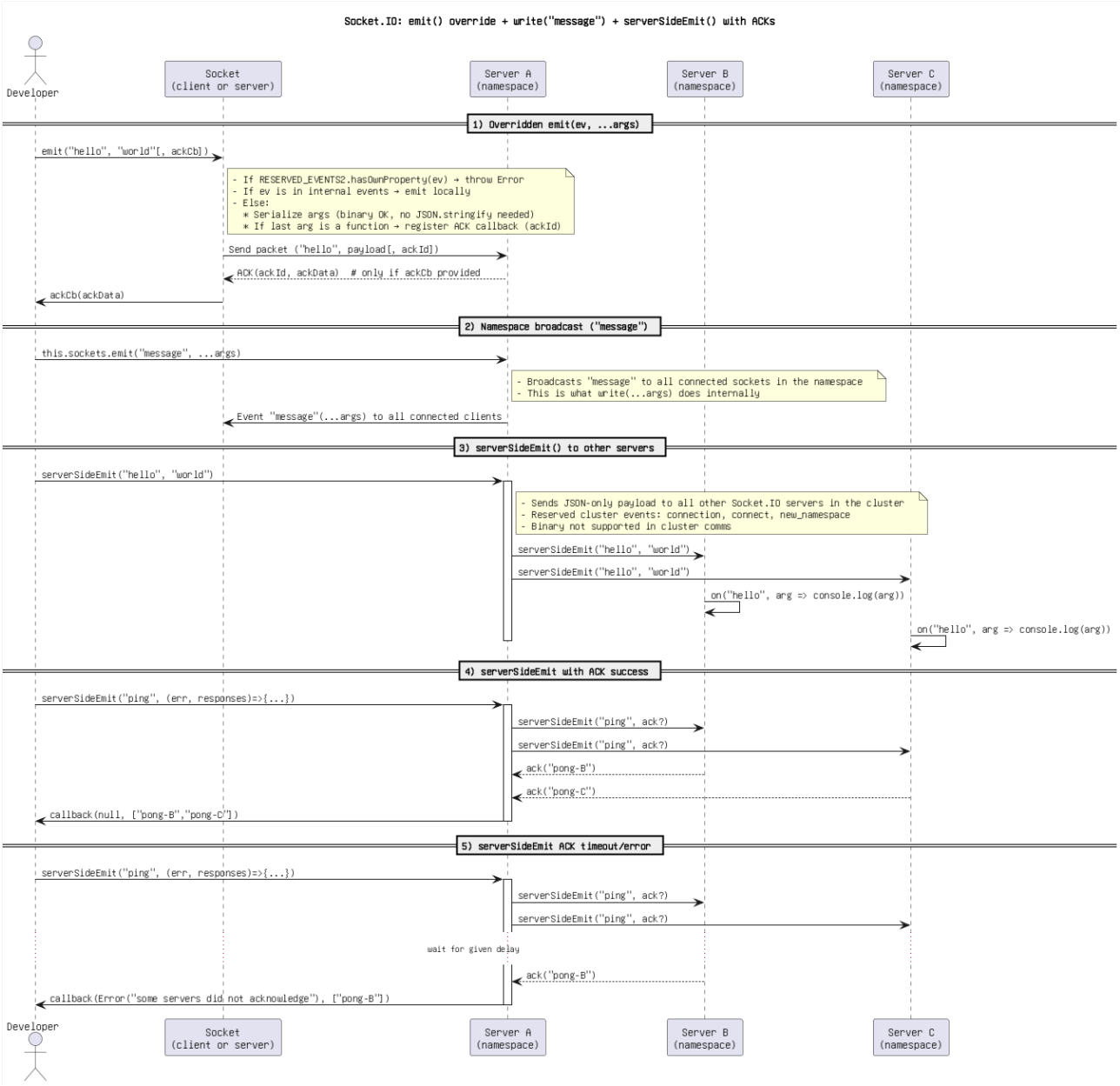


Figure 36: Socket.IO_ emit() override + write(_message_) + serverSideEmit() with ACKs(3).png



gure 37: Socket.IO_ emit() override + write(_message_) + serverSideEmit() with ACKs(4) (Copy).pn

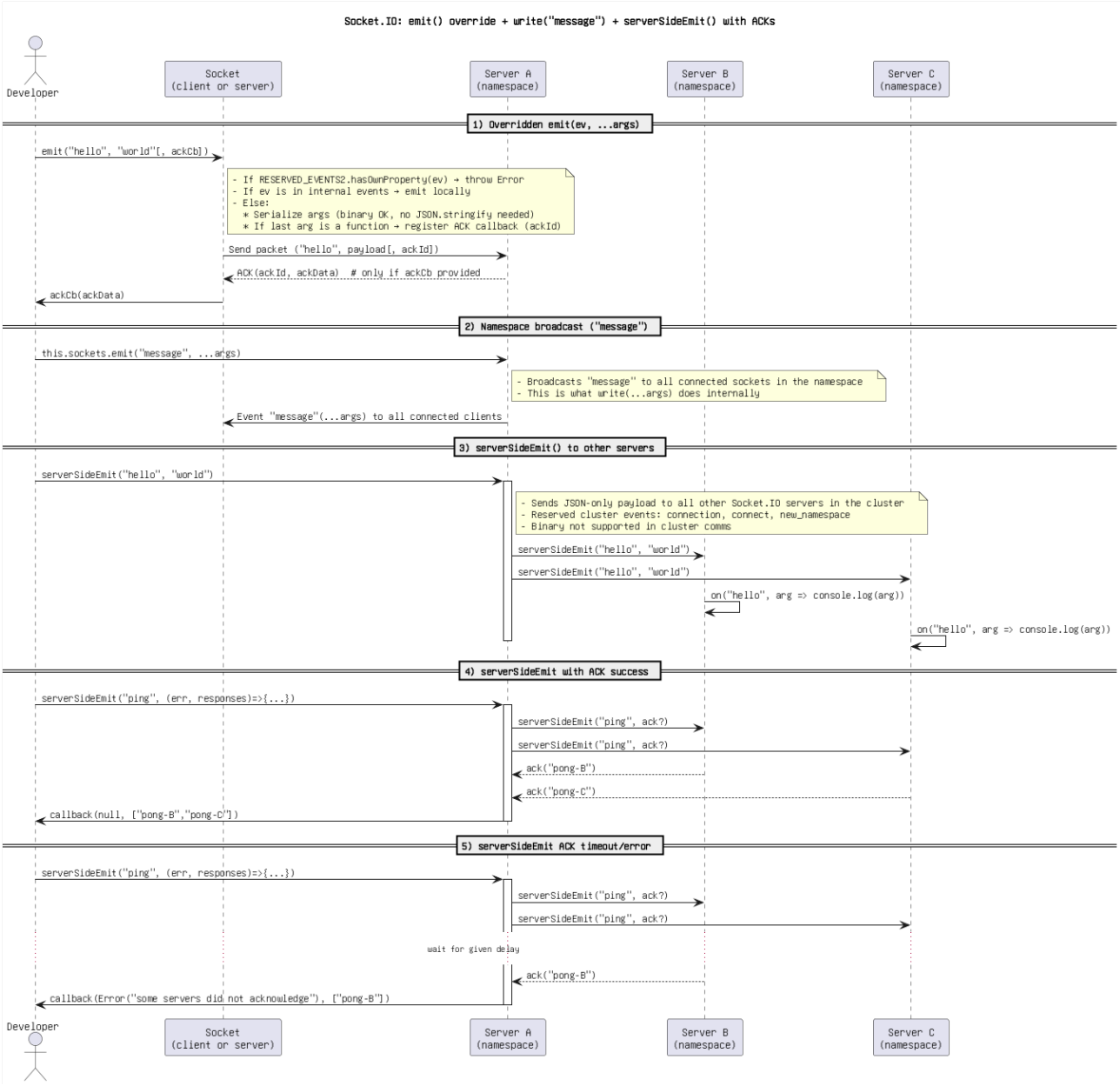


Figure 38: Socket.IO_ emit() override + write(_message_) + serverSideEmit() with ACKs(4).png

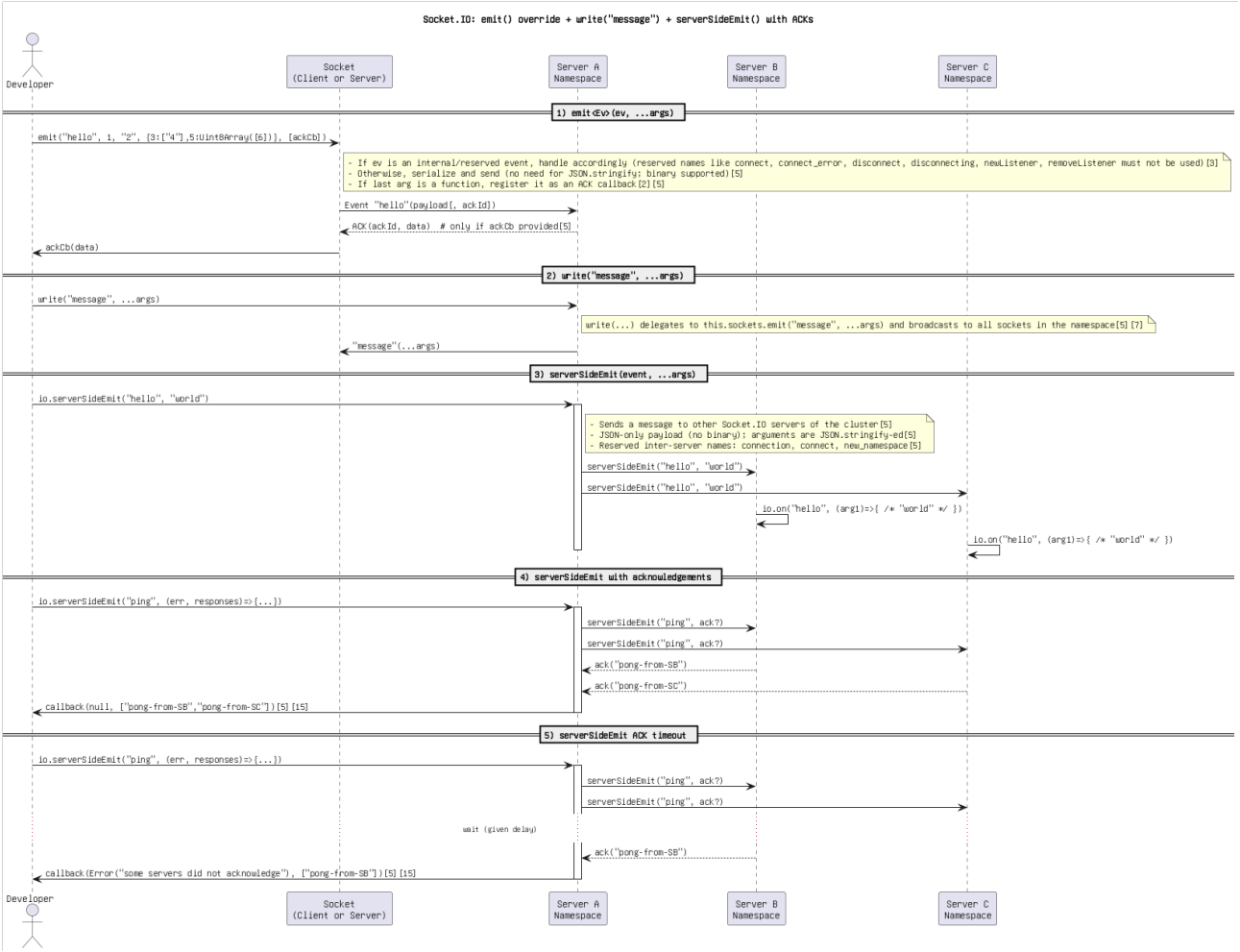


Figure 39: Socket.IO_ emit() override + write(_message_) + serverSideEmit() with ACKs..png

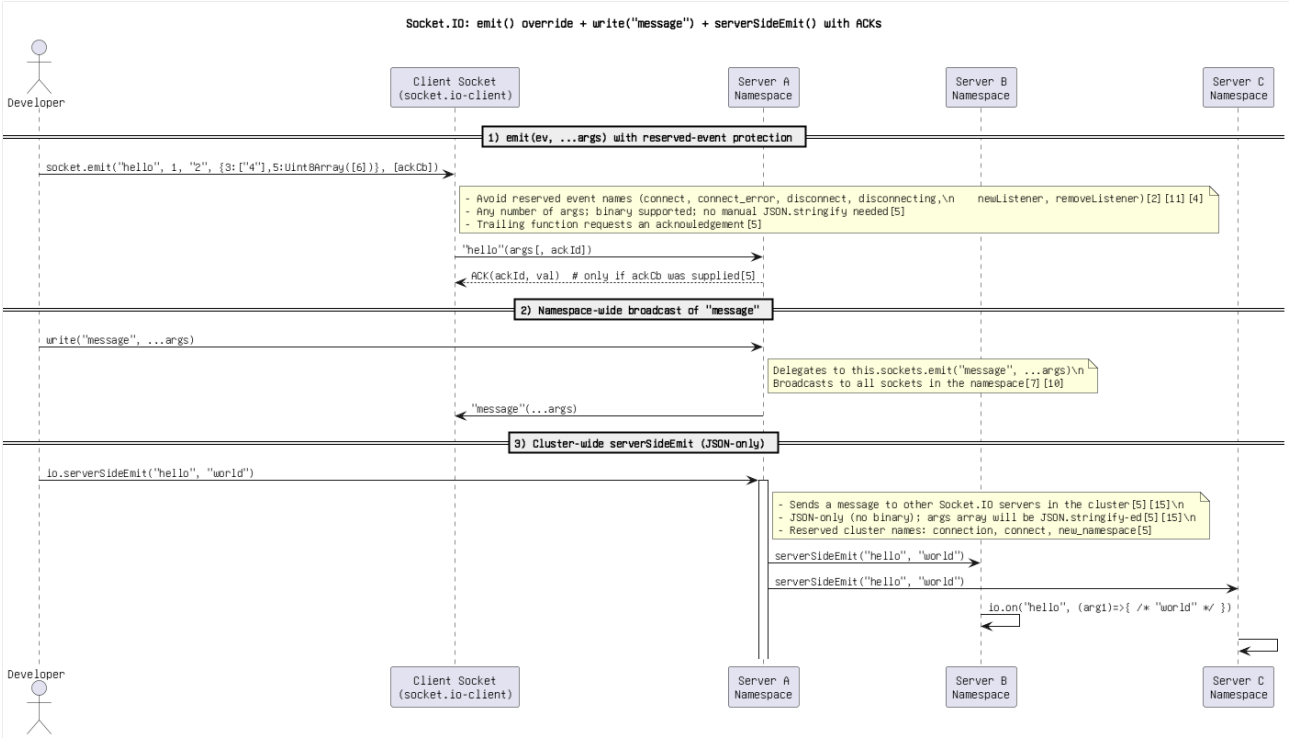


Figure 40: Socket.IO_ emit() override + write(_message_) + serverSideEmit() with ACKs.png

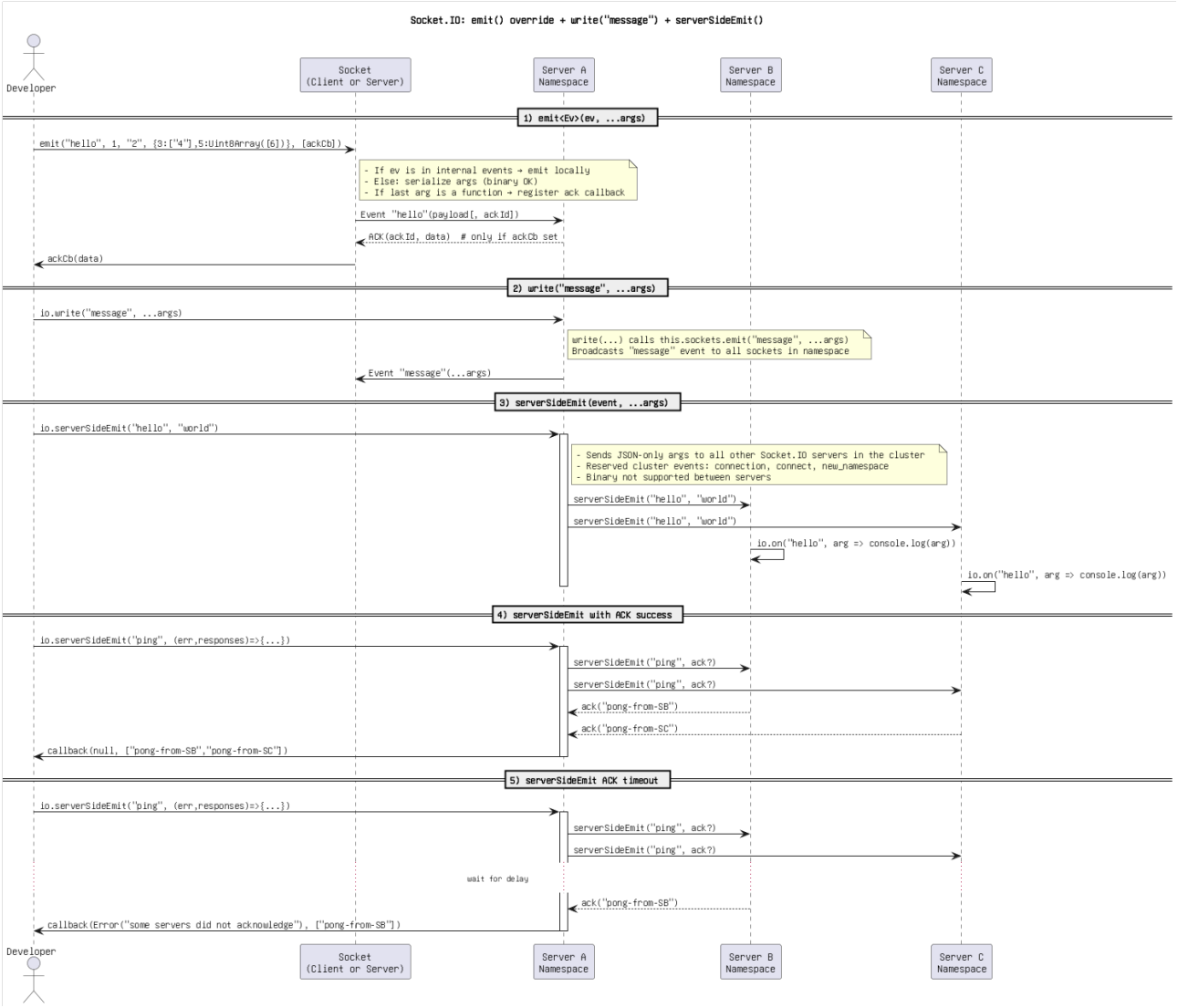


Figure 41: Socket.IO_ emit() override + write(_message_) + serverSideEmit()(1).png

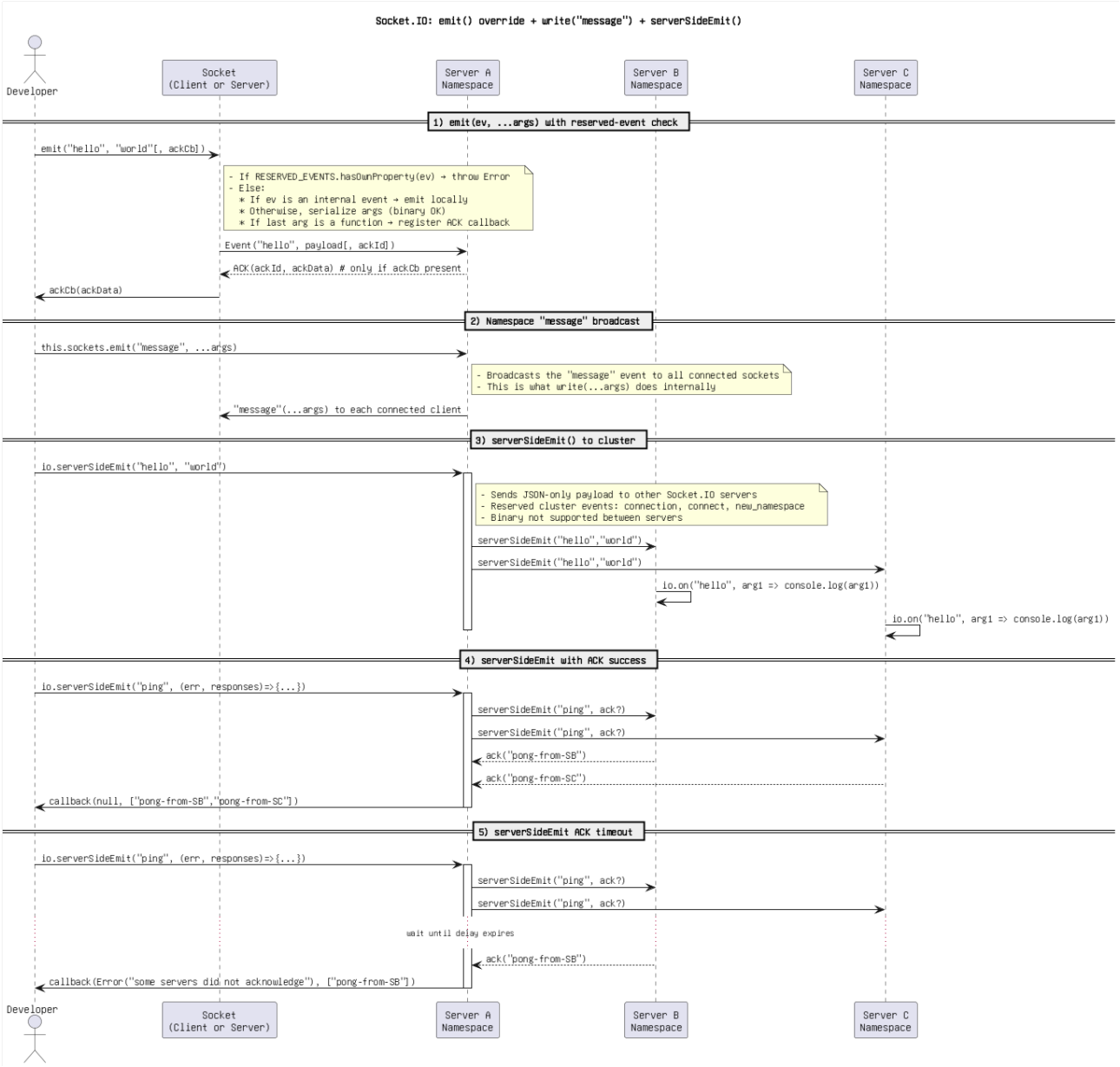


Figure 42: Socket.IO_ emit() override + write(_message_) + serverSideEmit()(2).png

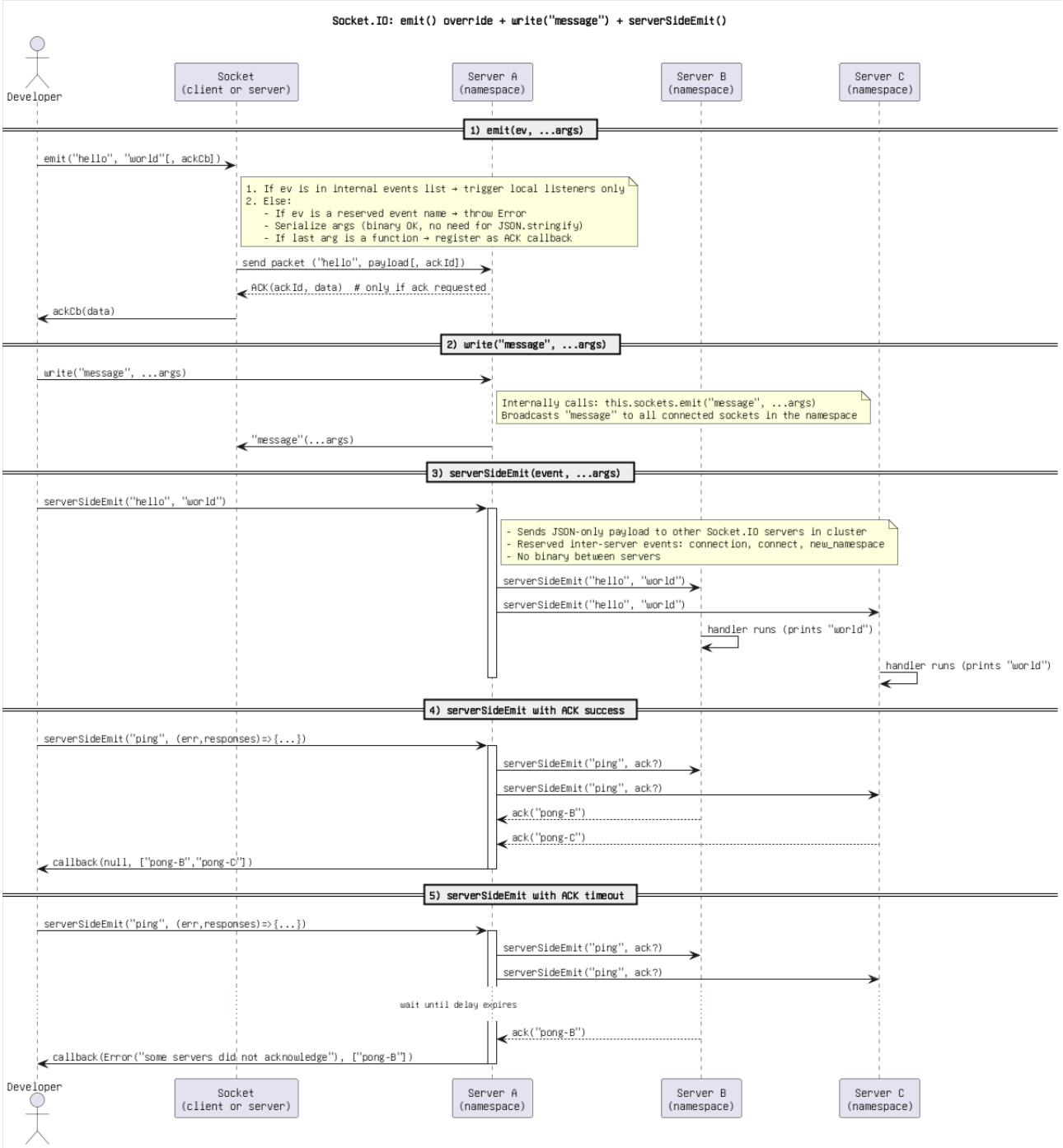


Figure 43: Socket.IO_ emit() override + write(_message_) + serverSideEmit()(3).png

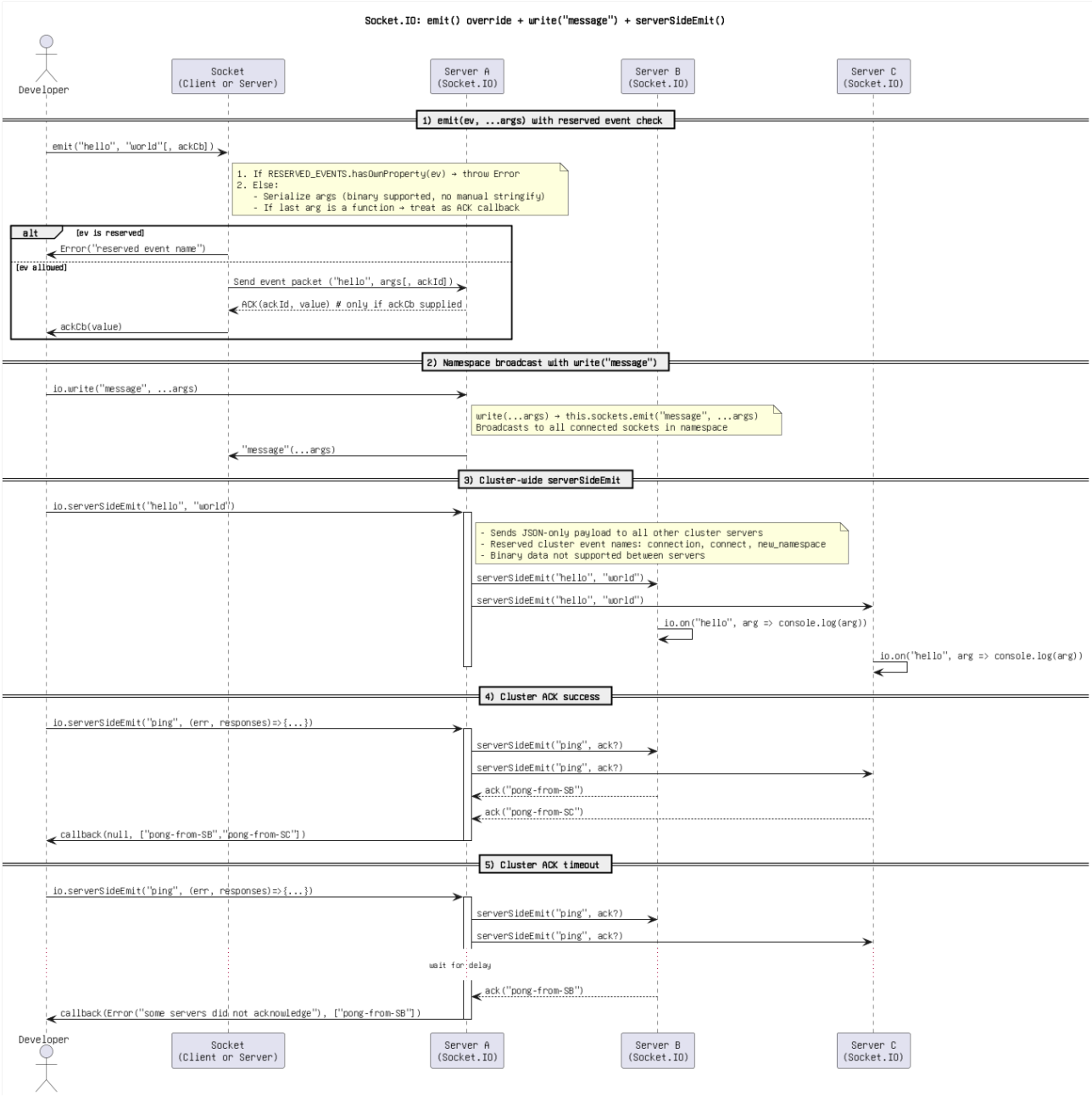


Figure 44: Socket.IO_ emit() override + write(_message_) + serverSideEmit().png

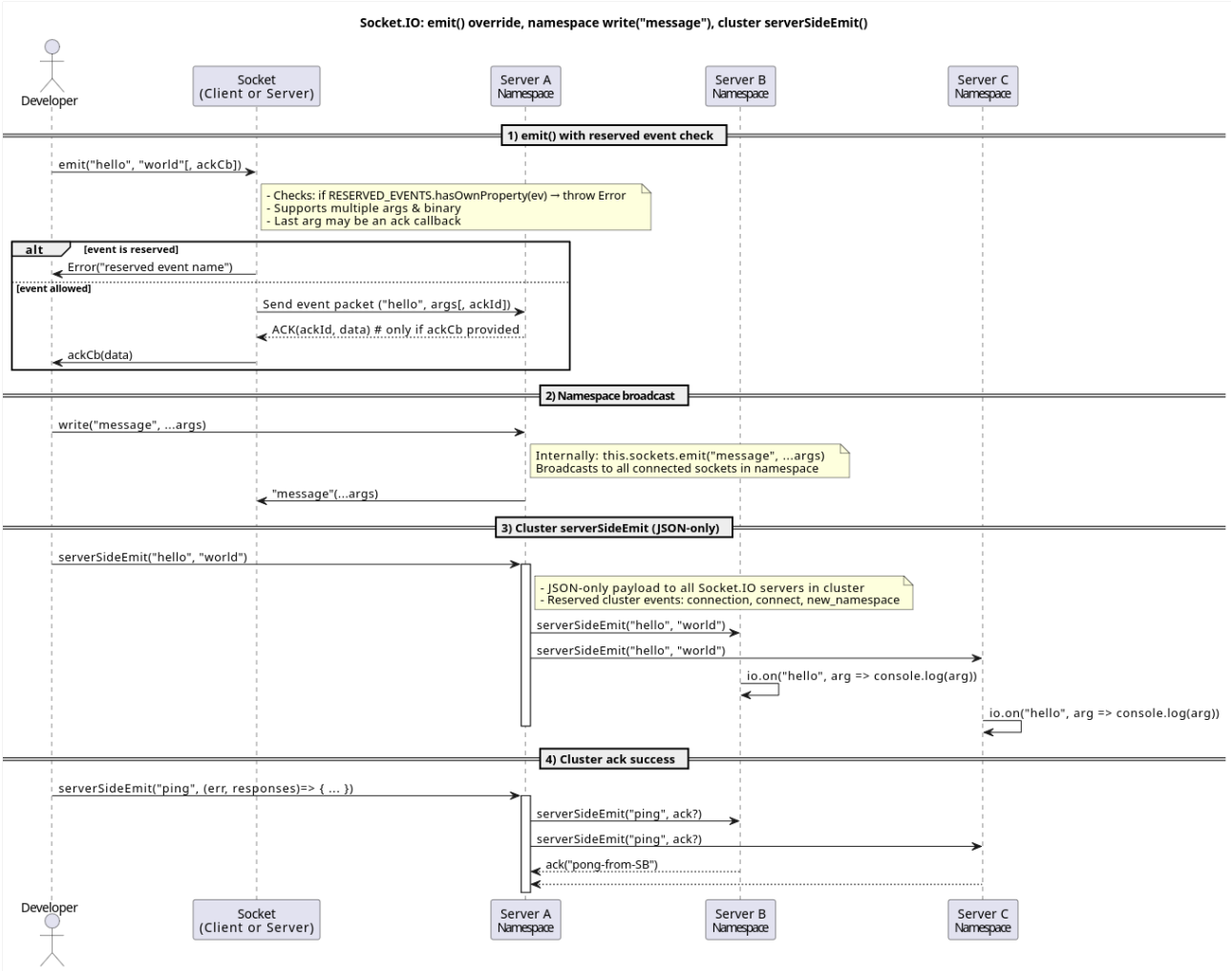


Figure 45: Socket.IO_ emit() override, namespace write(_message_), cluster serverSideEmit().png

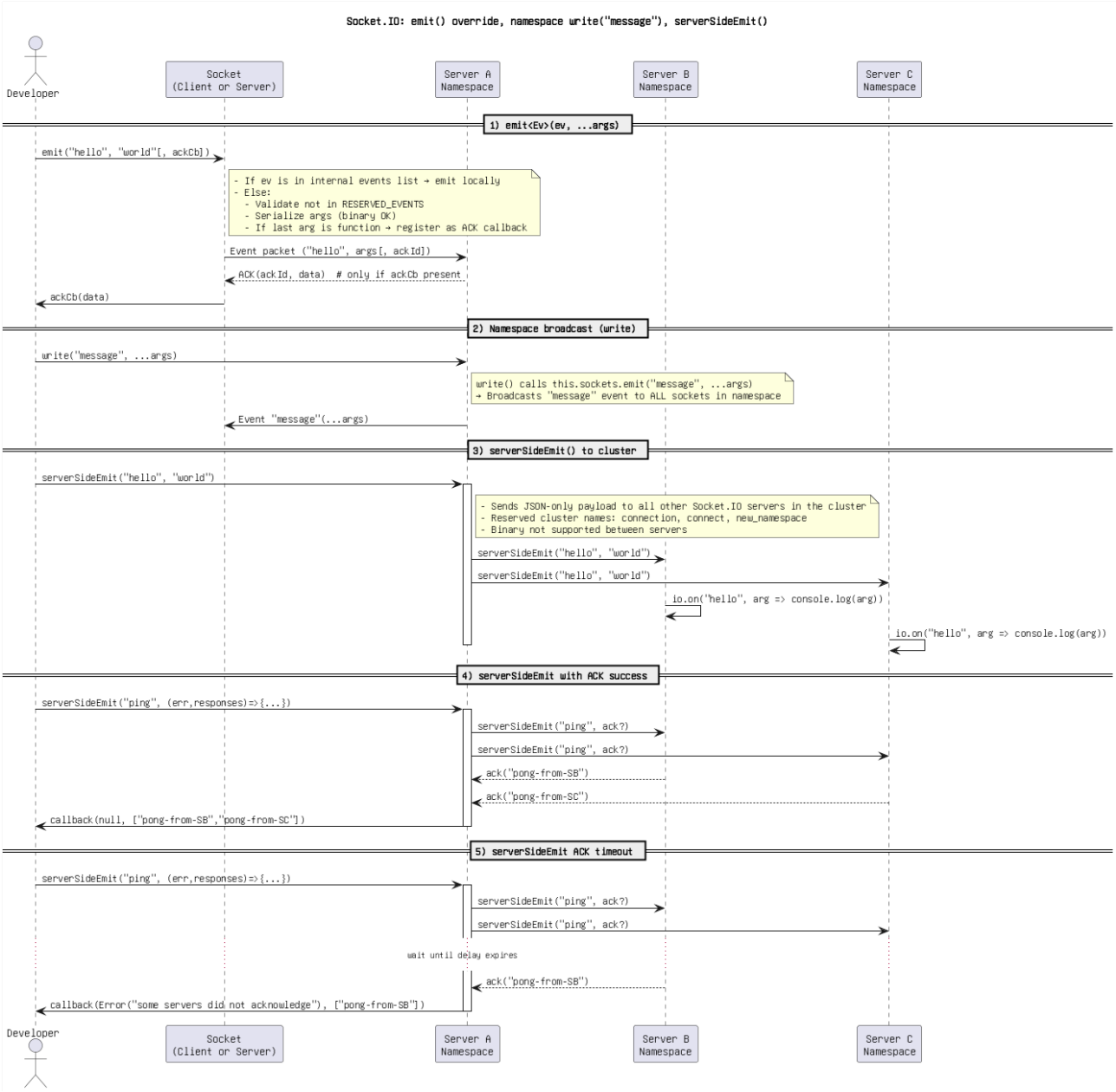


Figure 46: Socket.IO_ emit() override, namespace write(_message_), serverSideEmit().png

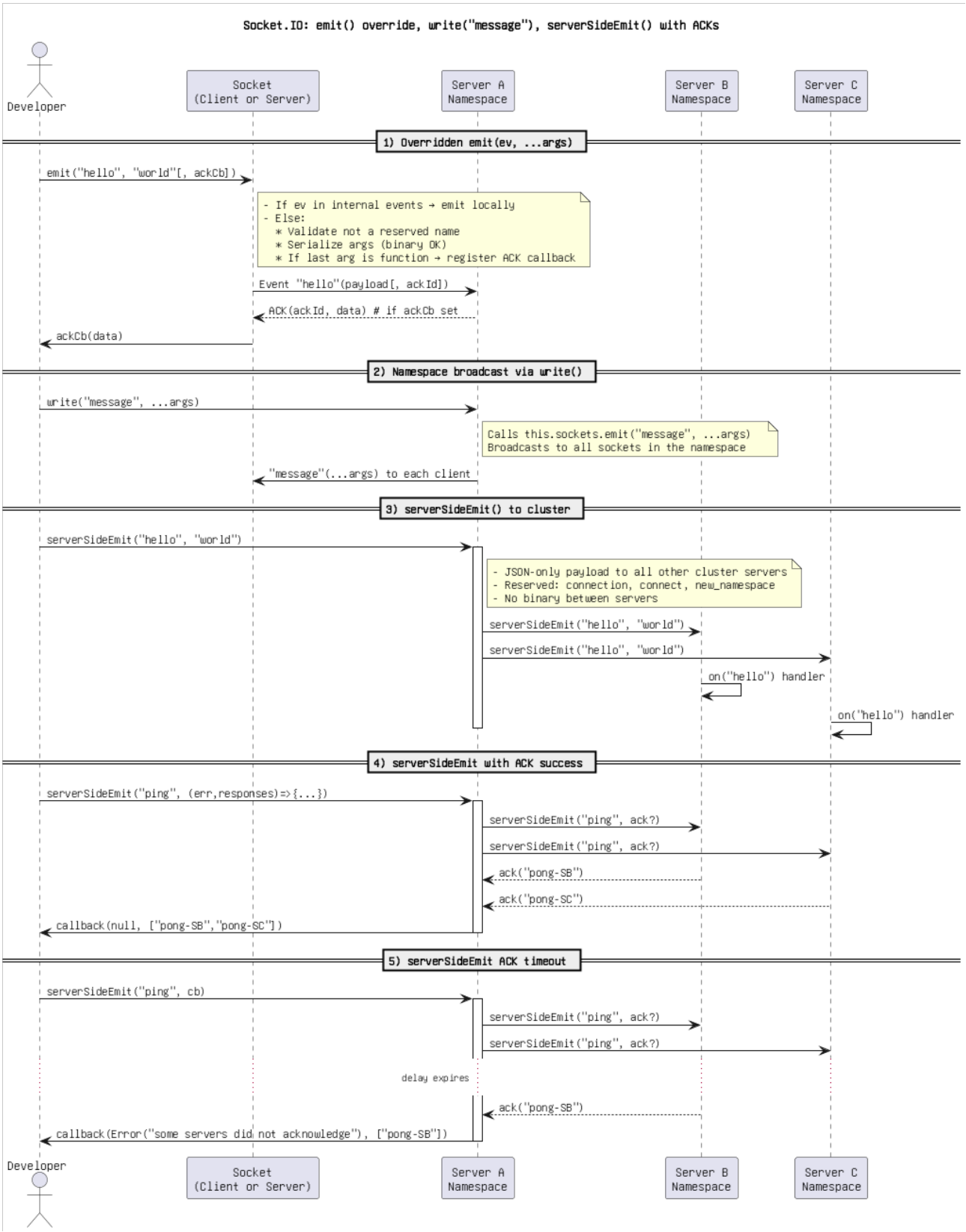


Figure 47: Socket.IO_ emit() override, write(_message_), serverSideEmit() with ACKs.png

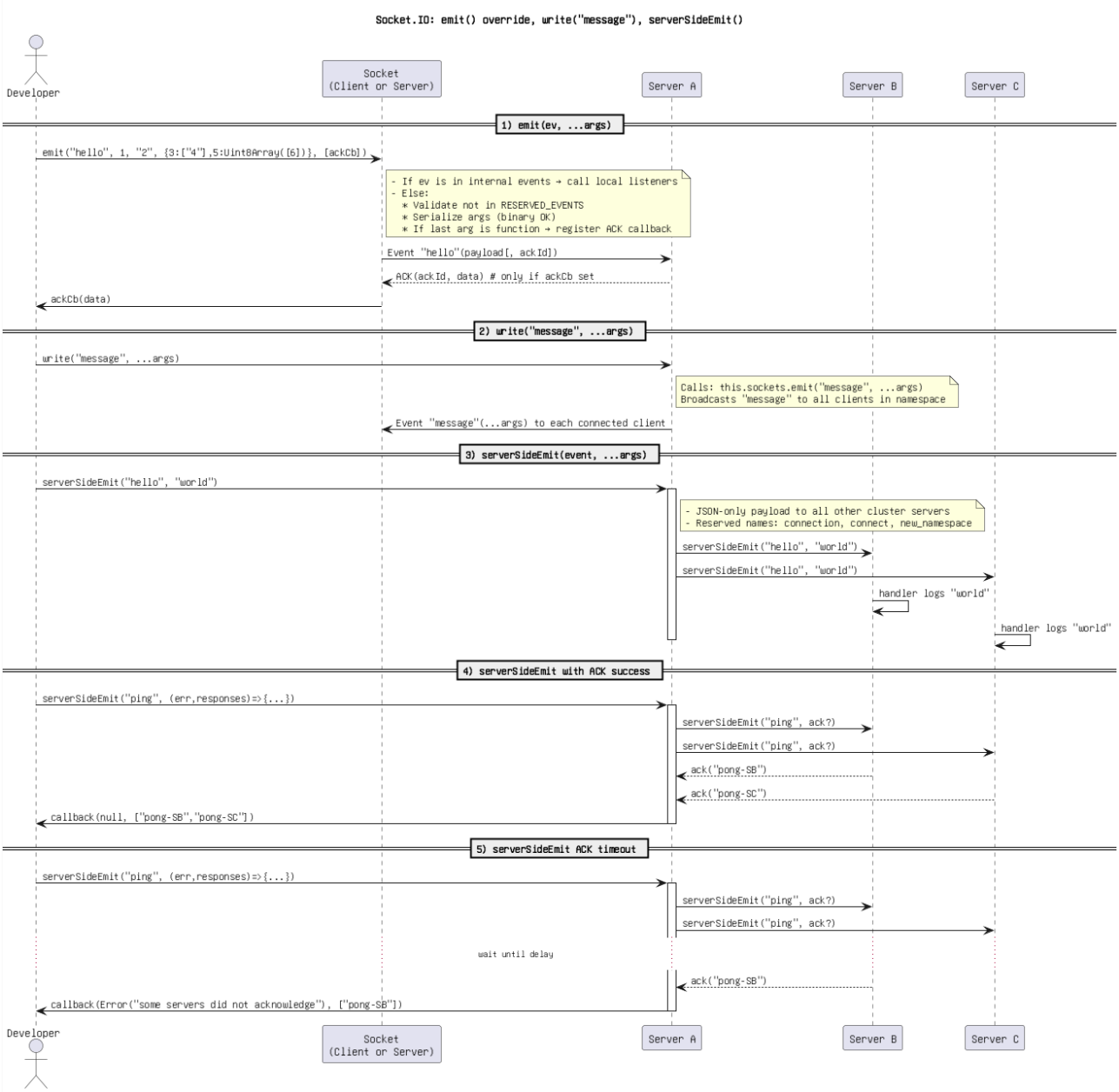


Figure 48: Socket.IO_ emit() override, write(_message_), serverSideEmit().png

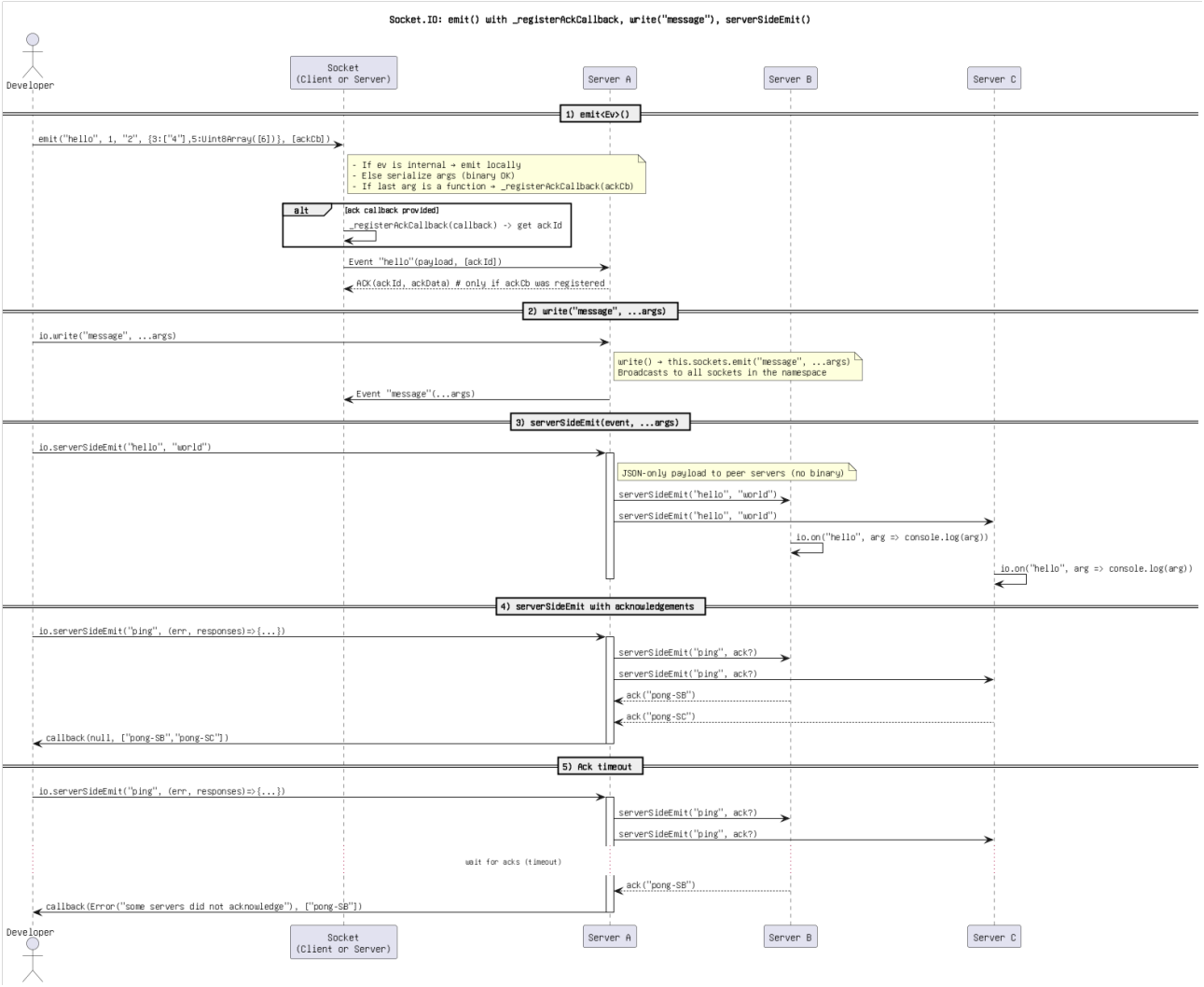


Figure 49: Socket.IO_ emit() with _registerAckCallback, write(_message_), serverSideEmit().png

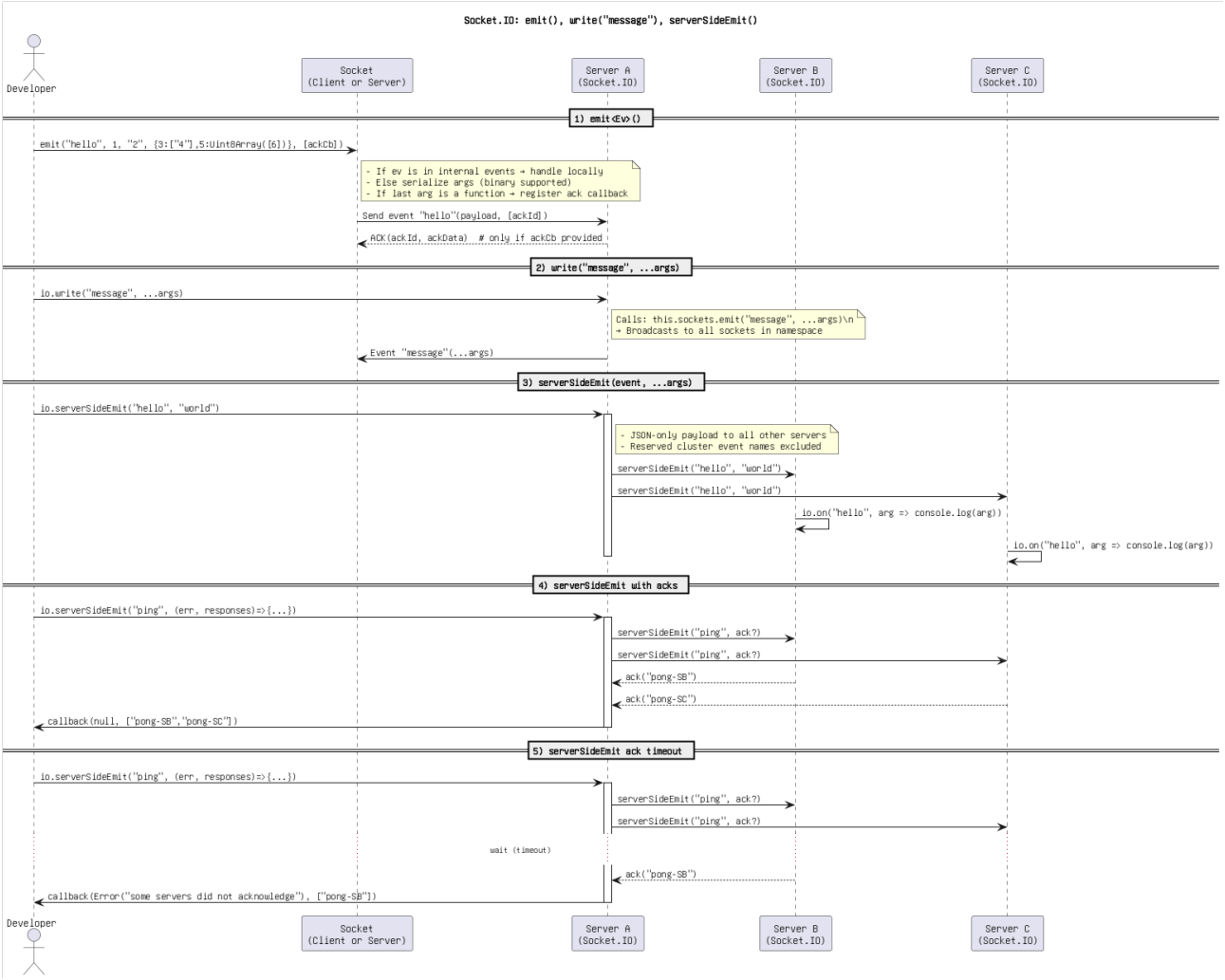


Figure 50: Socket.IO_ emit(), write(_message_), serverSideEmit().png

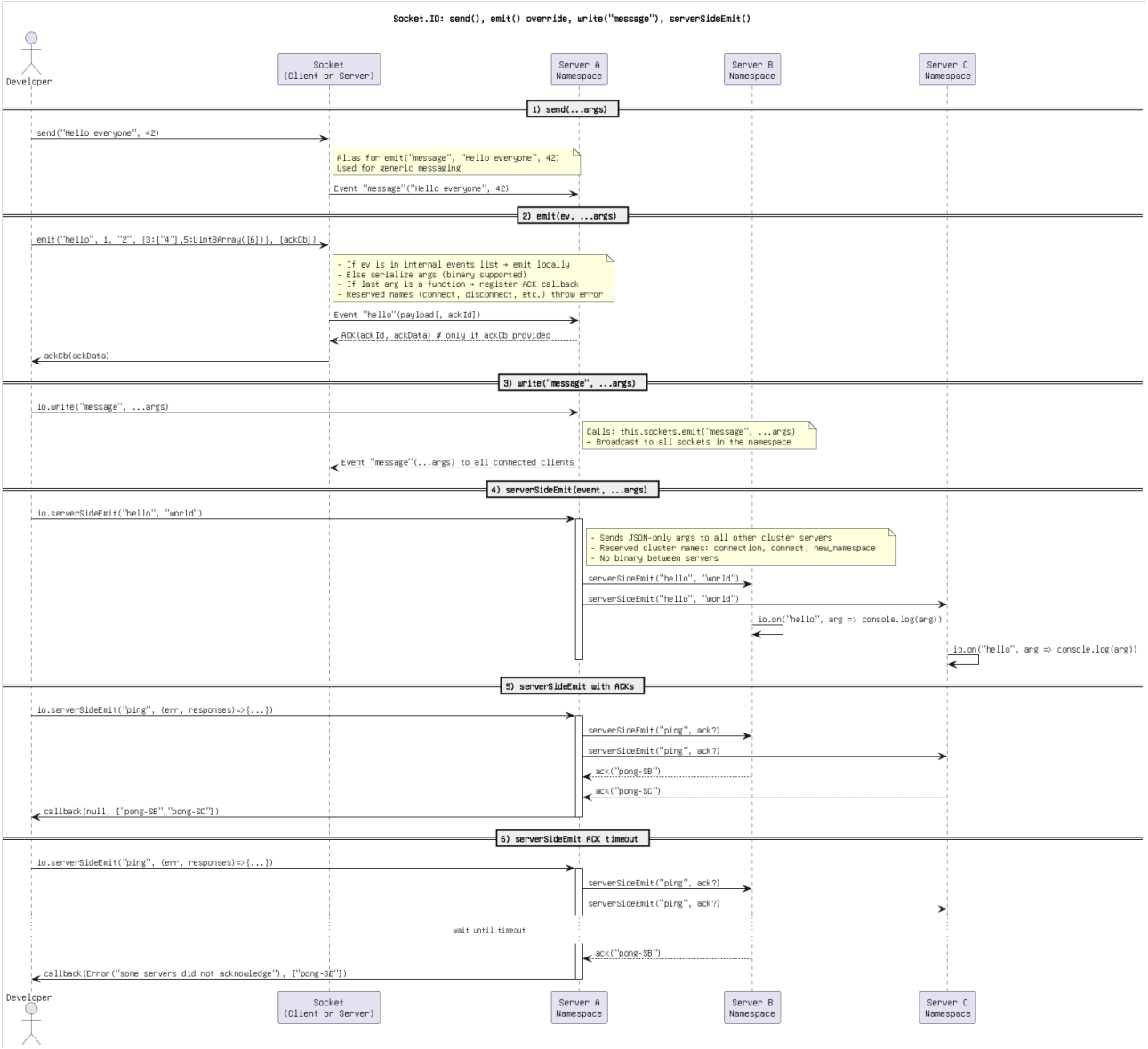


Figure 51: Socket.IO_ send(), emit() override, write(_message_), serverSideEmit().png

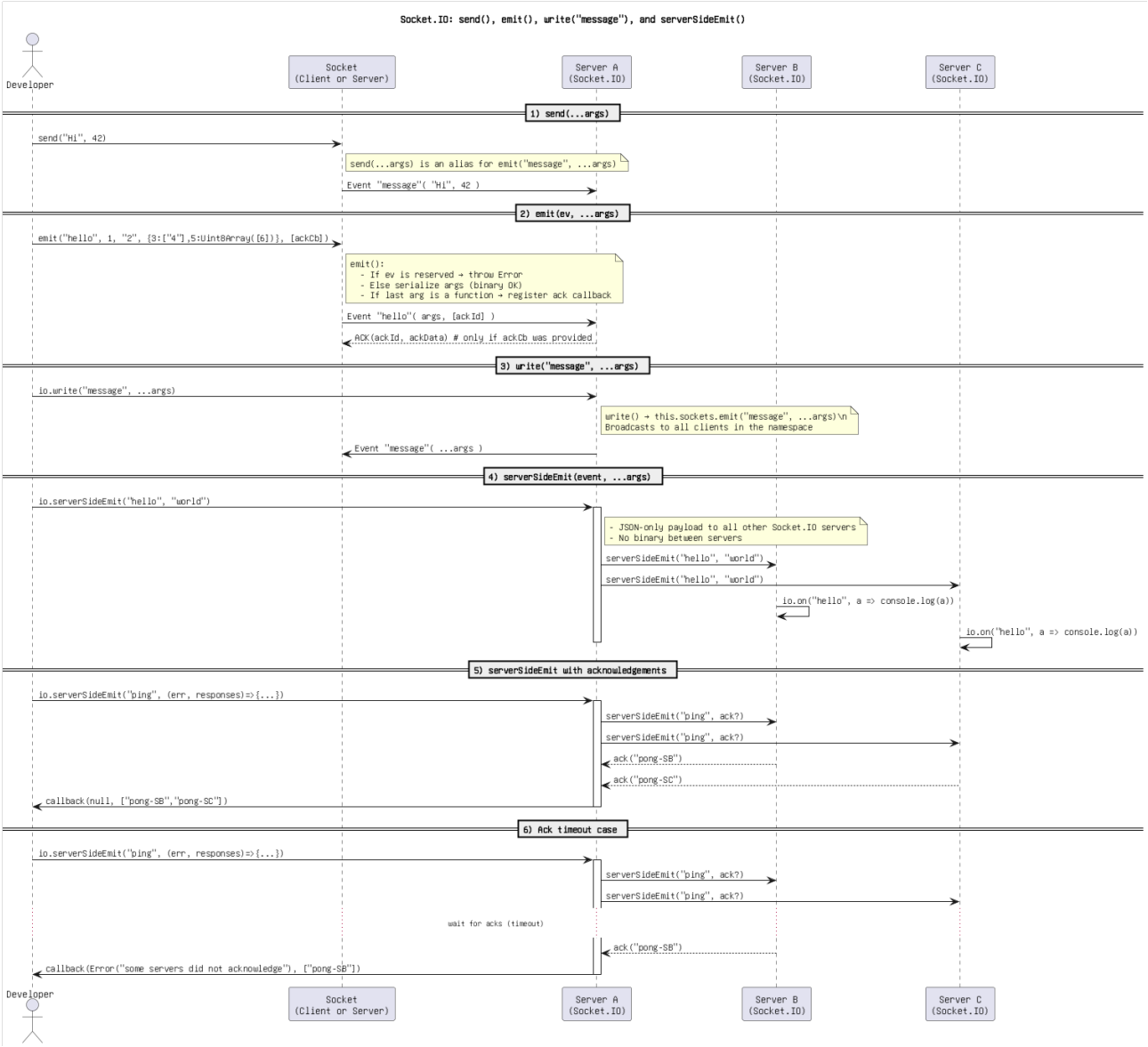


Figure 52: Socket.IO_ send(), emit(), write(_message_), and serverSideEmit().png

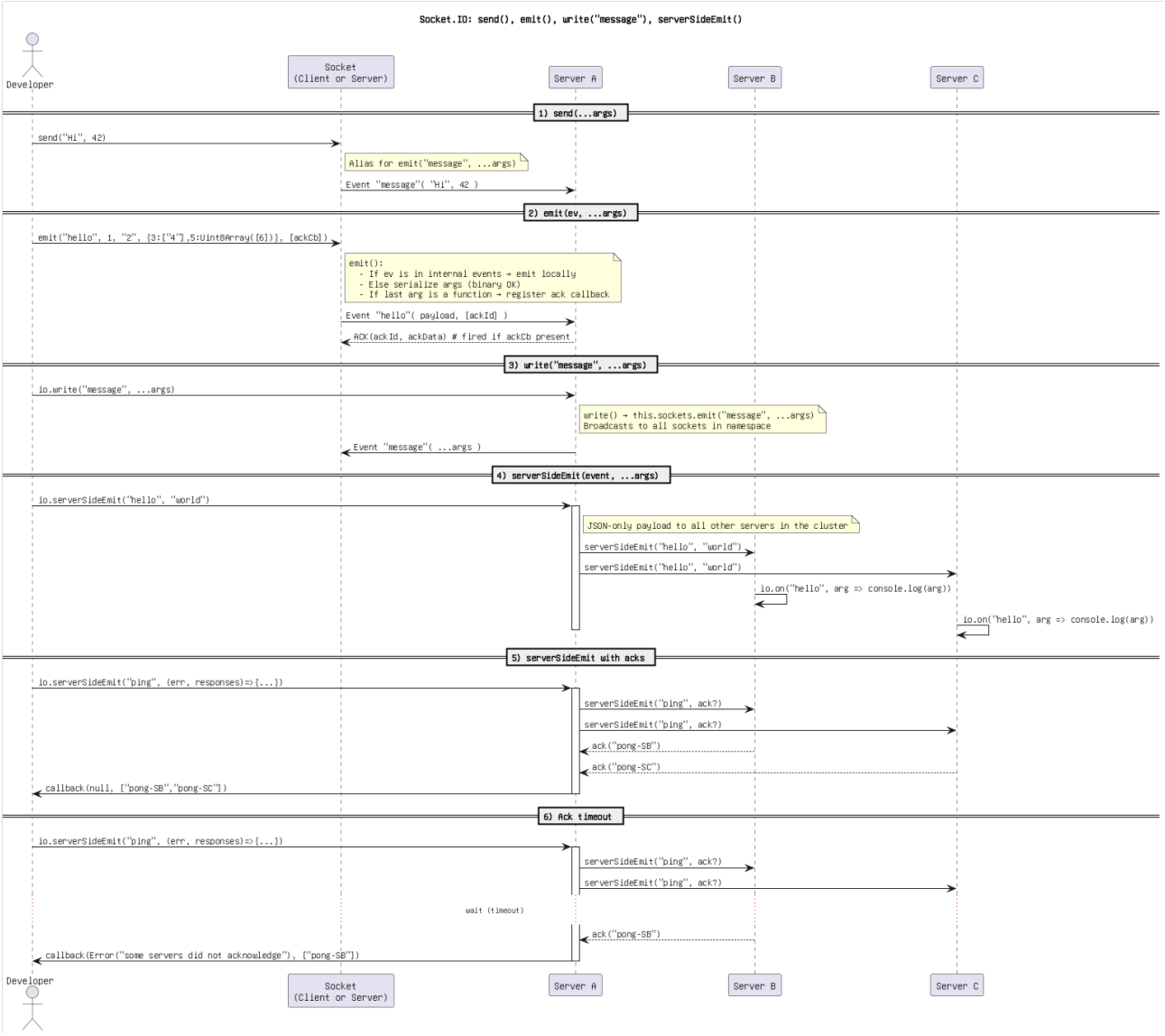


Figure 53: Socket.IO_ send(), emit(), write(_message_), serverSideEmit().png

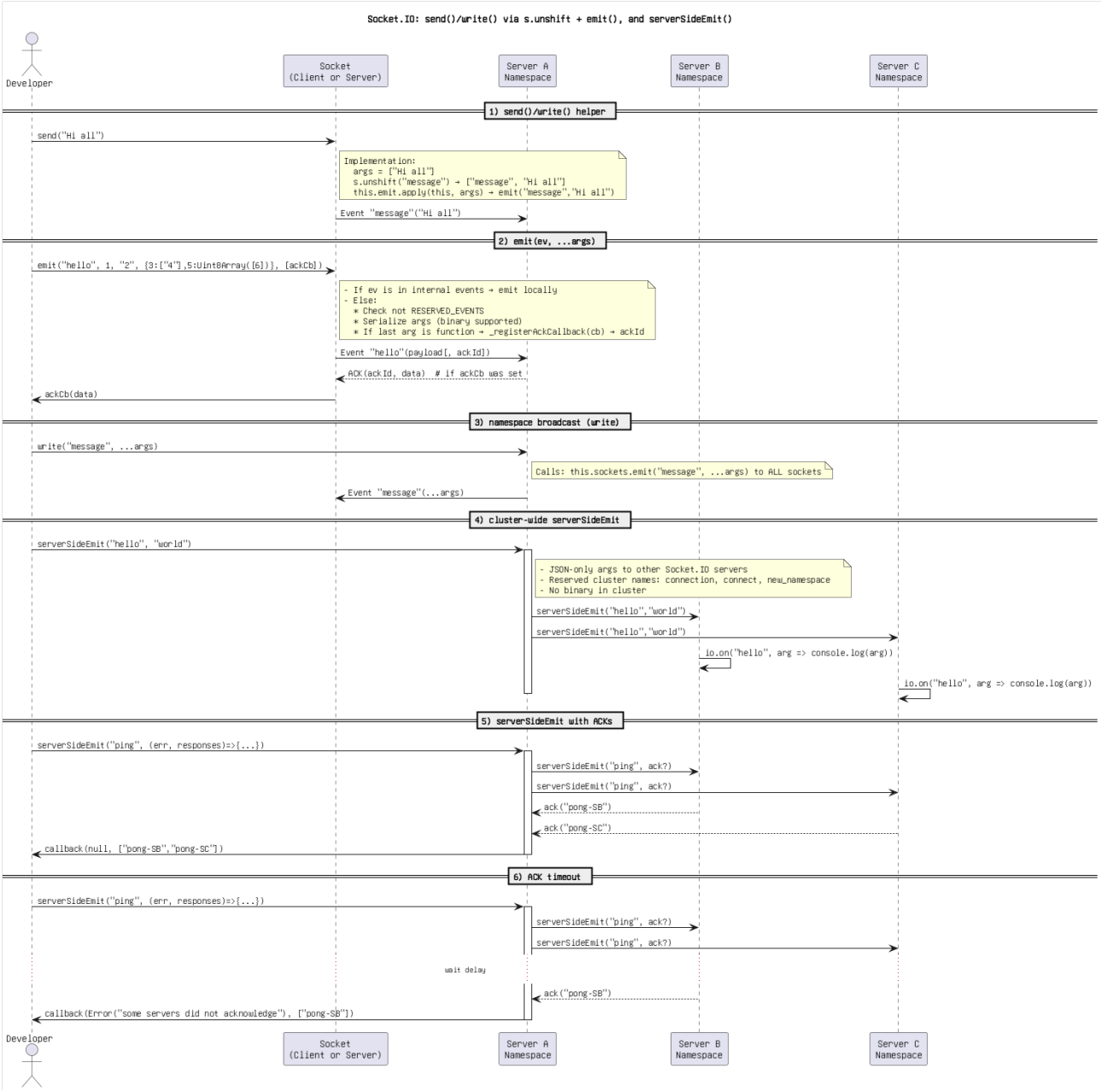
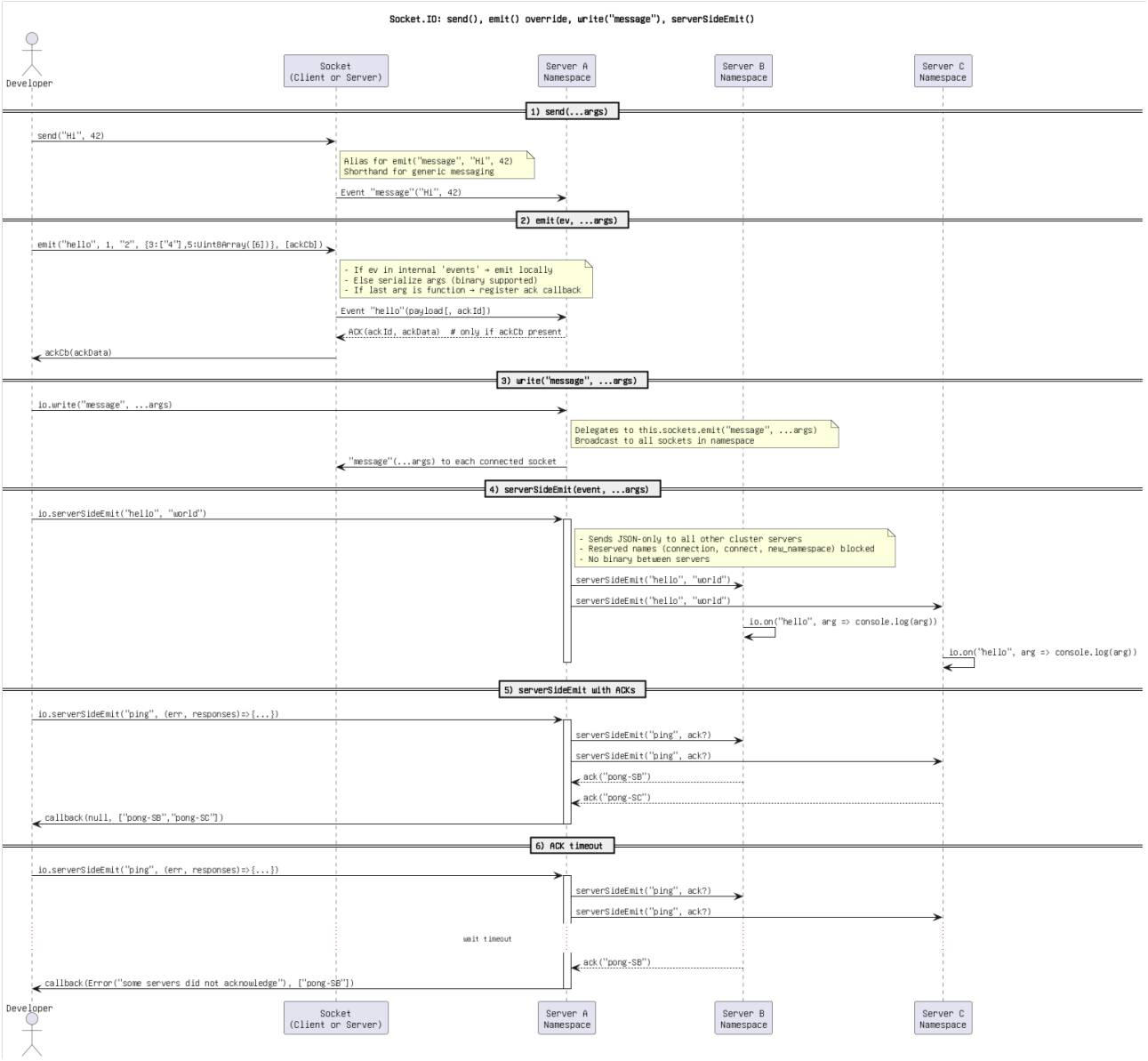


Figure 54: Socket.IO_ send()_write() via s.unshift + emit(), and serverSideEmit().png



gure 55: Untitled DiSocket.IO_ send(), emit() override, write(_message_), serverSideEmit()agram.pr

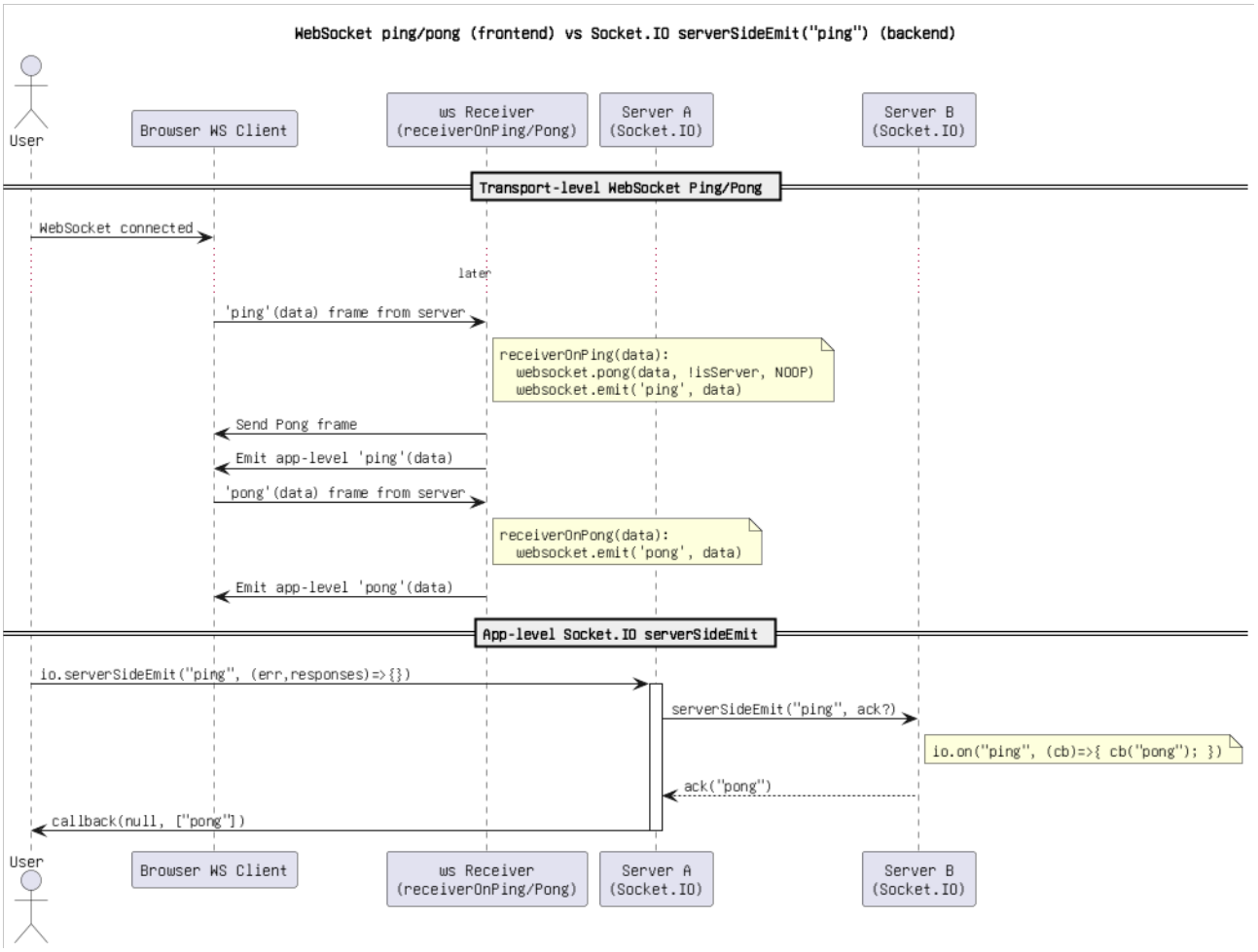
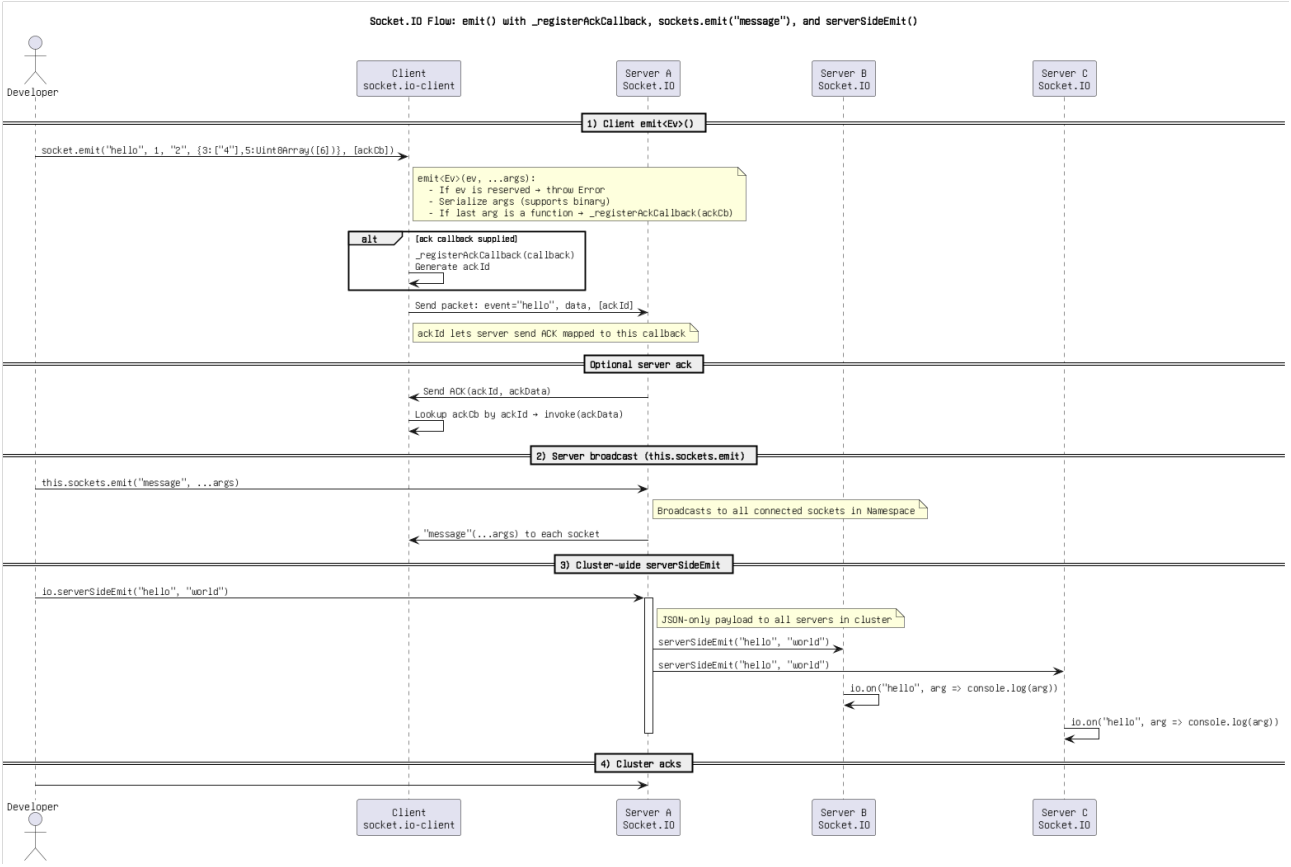
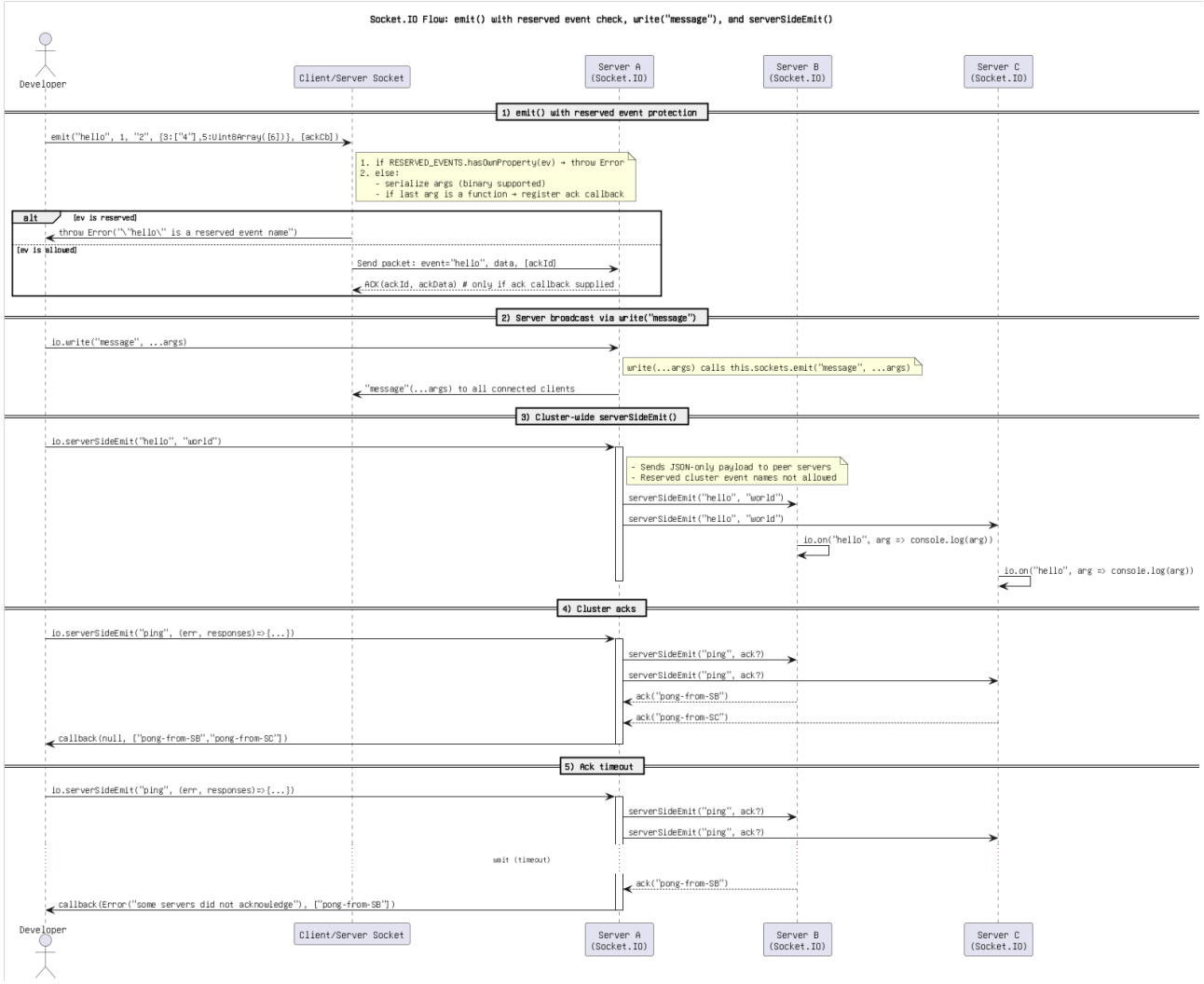


Figure 56: WebSocket ping_pong (frontend) vs Socket.IO serverSideEmit(_ping_) (backend).png



57: socket.IO Flow_ emit() with _registerAckCallback, sockets.emit(_message_), and serverSideEmi



ure 58: socket.IO Flow_ emit() with reserved event check, write(_message_), and serverSideEmit().p