Print this page to PDF (Ctrl/Cmd+P) Print to PDF [Full case study]

Sevendyne
Engineering firm · Since 2016
Technical one-pager

Tacodi — SIP & conference telephony (C++)

Engagement: Native C++ on client product team **Stack:** C++ · SIP · VoIP · conference bridges **Domain:** Live conferencing / telecom

Low-level C++ for SIP session lifecycle, PSTN/VoIP bridging, and network phone interfaces — the plumbing behind operator-facing conference products, not a web wrapper.

C++SIPVoIPConference bridgesNative telecom
Architecture

Session control vs media path

SIP signalling (INVITE/ACK/BYE, re-INVITE for hold/transfer) runs on dedicated threads separate from media mixing. Phone-interface adapters normalize hardware events into the same state machine conference cores use for soft clients.

Design decisions

Why native C++

Decision	Rejected	Why
Native C++ modules	Node/Java SIP stacks only	Latency + direct hardware/codec integration Tacodi already had
Explicit call FSM	Ad-hoc flags per feature	Conference transfers and hold states stay auditable under load
Interface adapters per phone type	One mega driver	Field deployments vary; adapters isolate vendor quirks

Reliability

Production constraints

- Graceful teardown on partial network loss — no zombie SIP dialogs
- Logging tied to `Call-ID` for field support correlation
- Release cadence aligned to Tacodi field feedback, not agency milestones

Outcomes

Results

- Production SIP and phone-interface modules in conferencing products
- Proof of native C++ depth alongside Speechlogix telecom lineage
- Comparable complexity to Trading Qt — different domain, same engineering bar

Tacodi programme. Details available under NDA.

contact@sevendyne.com · sevendyne.com/case-studies