

CASE STUDY

Transform WebRTC Media Server & Application into Agentic AI Platform



Many organizations that adopted WebRTC over the last decade built reliable conferencing platforms around mature media servers such as Janus, Jitsi, Kurento, or FreeSWITCH. Today, these organizations want to introduce Agentic AI capabilities—AI meeting assistants, digital avatars, voice agents, real-time translation, meeting summaries, and conversational AI.

The common recommendation is often to migrate to a newer media server ecosystem that offers built-in AI integration. However, such migrations are expensive, risky, and frequently unnecessary.

SpringCT recently helped a customer preserve their existing investment by integrating Agentic AI into a production conferencing platform built on Janus Media Server—without replacing the core media infrastructure. Instead, SpringCT designed and implemented an intelligent media bridge between Janus and LiveKit, allowing both platforms to coexist seamlessly.

The result was a significantly lower implementation cost, minimal business disruption, and a future-ready architecture capable of supporting AI-driven experiences.

Customer Challenge

The customer had already invested several years in building a conferencing platform around Janus Media Server. The platform included:

- Multi-party conferencing
- Screen sharing
- Recording
- SIP interoperability
- User authentication
- Meeting management
- Custom moderation features
- Mobile and web applications
- Operational monitoring and deployment automation

As the customer planned to introduce an AI-powered avatar service, they discovered that the third-party AI platform was tightly integrated with LiveKit for real-time media processing. The straightforward approach appeared to be replacing Janus entirely with LiveKit.

However, this raised several concerns:

- Rebuilding years of WebRTC development
- Rewriting media workflows
- Revalidating production stability
- Migrating thousands of lines of client code
- Re-testing every conferencing feature
- Increased deployment and operational risk
- Extended project timelines

The migration effort threatened to outweigh the actual AI implementation.

SpringCT's Evaluation

Rather than recommending a complete migration, SpringCT performed a detailed assessment of the existing platform.

The evaluation focused on:

- Existing media architecture
- WebRTC signaling flows
- Publisher/subscriber management
- Recording workflows
- SIP integration
- Conference control APIs
- Client SDK dependencies
- Scalability requirements
- AI integration requirements
- Latency expectations

The assessment revealed that over 90% of the conferencing platform remained fully capable of supporting future business needs. The only missing capability was seamless interoperability with AI services that expected LiveKit rooms. This insight changed the entire implementation strategy.

Features That Would Have Been Impacted by a Full Media Server Migration

Replacing the media server would have required extensive redevelopment across multiple layers.

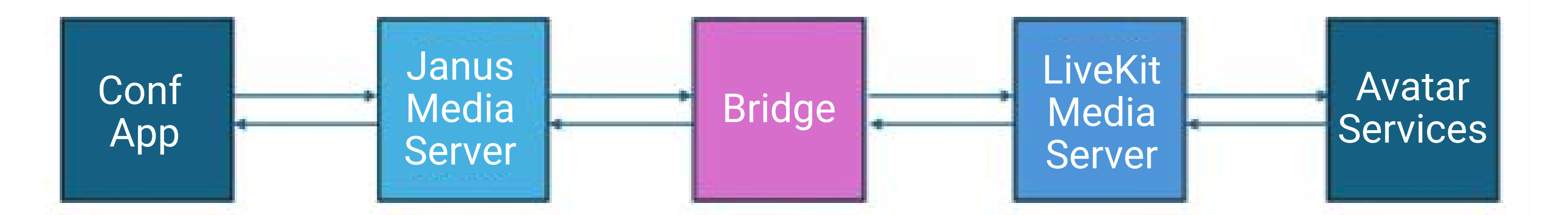
- **Media Layer:** Publisher management , Subscriber handling , Simulcast configuration, Bandwidth adaptation, Recording pipeline
- **Client Applications:** Web SDK changes, Mobile SDK changes, Room lifecycle, Device management , ICE and signaling changes
- **Server Components:** Conference orchestration, User presence, Media event handling, Monitoring, Analytics
- **Operational Components:** Deployment automation, Scaling policies, Production monitoring, Existing troubleshooting tools

Each of these changes would require development, QA, production rollout, and user validation.

SpringCT's Solution

Instead of replacing Janus, SpringCT introduced a **Janus–LiveKit Media Bridge**.

The bridge allowed both media servers to operate together while serving different responsibilities.



Product Features

- **Stream Detection:** Detecting newly published audio/video streams.
- **Publisher Subscription:** Subscribing to Janus publishers.
- **Media Forwarding:** Forwarding media streams into corresponding LiveKit rooms.
- **Participant Lifecycle Synchronization:** Synchronizing participant lifecycle events.
- **Low-Latency Media Transport:** Maintaining low-latency media transport.
- **AI Media Reception:** Receiving AI-generated media from LiveKit.
- **AI Media Republishing:** Republishing AI audio/video back into Janus conferences.
- **Stream Cleanup:** Managing stream cleanup when participants left.

Key Technical Outcomes

- **Janus-to-LiveKit real-time media interoperability**
- **LiveKit-to-Janus AI media publishing**
- **Transparent participant synchronization**
- **Low-latency media forwarding**
- **Support for third-party avatar services**
- **Minimal changes to the existing application**
- **Future-ready architecture for additional Agentic AI capabilities**

Business Benefits

The bridge-based architecture delivered several important advantages.

- **Preserved Existing Investment:** The customer retained years of engineering investment without rewriting the conferencing platform.
- **Faster AI Adoption:** Agentic AI capabilities were introduced without waiting for a large-scale platform migration.
- **Lower Project Risk:** Critical conferencing features remained untouched, reducing regression risk.
- **Incremental Modernization:** LiveKit could be introduced gradually where required rather than replacing the entire infrastructure.
- **Production Stability:** Existing customers experienced no disruption while AI features were added.
- **Vendor Flexibility:** The architecture remained open for future AI providers without locking the conferencing platform into a single media server.

Estimated Cost Savings

A complete migration from one production media server to another typically involves:

- Platform redesign
- Client SDK migration
- Feature redevelopment
- Extensive QA
- Production rollout
- User acceptance testing
- Operational retraining

For mature conferencing platforms, these efforts can translate into tens of thousands of dollars in engineering costs, with significantly higher costs for enterprise deployments.

By introducing a targeted interoperability layer instead of replacing the media infrastructure, SpringCT enabled the customer to preserve their existing investment while accelerating AI adoption.

Conclusion

- Modern AI capabilities should not require organizations to discard years of successful WebRTC engineering.
- In many cases, the existing conferencing platform remains highly capable, requiring only a carefully designed interoperability layer to connect with emerging AI ecosystems.
- By developing a real-time bridge between Janus Media Server and LiveKit, SpringCT enabled the customer to integrate Agentic AI, digital avatars, and real-time conversational experiences while preserving their existing conferencing platform.
- The project demonstrates that modernization does not always require replacement. Sometimes, the most effective path forward is building the right bridge between proven technology and the next generation of AI.