



Nexus Health Systems

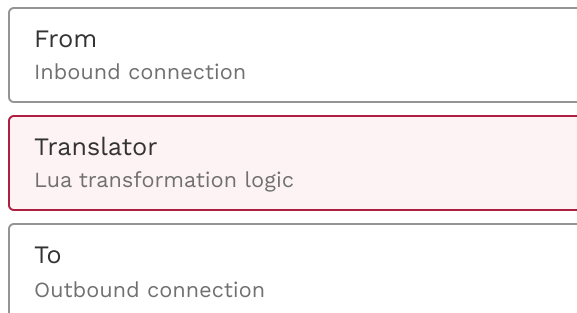
Migrating from Iguana to QIE without rebuilding every interface

Nexus Health Systems moved its production interfaces from INTERFACEWARE Iguana to the Qvera Interface Engine (QIE) without rewriting them and without interrupting message flow, using QIE's built-in AI Companion to port its existing Lua scripts to JavaScript.

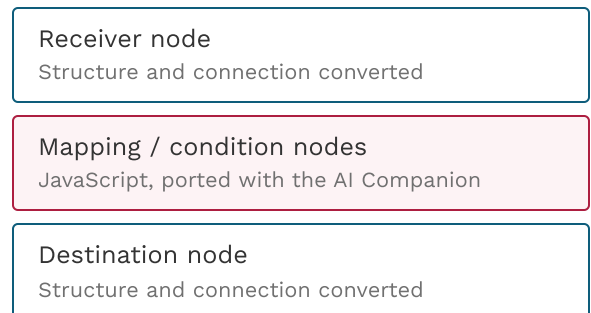
Changing interface engines can feel like a major undertaking. Existing integrations, custom scripts and years of accumulated logic make the idea of moving to a new platform look disruptive and slow. The work already invested is real, and nobody wants to spend it twice.

For Nexus Health Systems, a Houston-based health system, the move took less effort than the team expected — little enough that an integration project already on the schedule was never held up by it.

IGUANA



QIE



The middle row is the work. Everything above and below it converts.

HOW THE MIGRATION WORKS

The structure converts

Iguana's From / Translator / To maps onto QIE's receiver, mapping and destination nodes



The Lua is ported

The AI Companion rewrites the transformation logic as JavaScript, against QIE's own bindings



An engineer validates

A Qvera engineer reviews every converted interface with you before it goes live

Why did Nexus look at a new interface engine?

As Nexus evaluated the long-term direction of its interoperability environment, the team needed three things at once: flexibility in how and where it runs, real depth in healthcare integration, and the ability to move quickly.

Speed mattered more than it usually does. Nexus had an immediate integration project that needed to go forward, and a migration measured in quarters would have stalled it.

Four things shaped the decision:

1. **Licensing and deployment flexibility.** QIE runs on-premise, in containers, or in your own cloud on AWS, Azure or Google Cloud, and high availability is included rather than reserved for a top pricing tier.
2. **Security posture.** Qvera is SOC 2 Type 2 attested, and the attestation is independently audited.
3. **Healthcare interoperability experience.** QIE has been doing this work since 2008.
4. **The AI Companion built into the engine** — which turned out to matter most during the migration itself.

What happens to the Lua scripts?

This is the first question every Iguana site asks, and it was the largest single piece of work in the migration. Interface logic in Iguana is written in Lua; QIE scripts in JavaScript. Rather than rebuild the interfaces from scratch, Nexus used its existing Lua as the starting point. An Iguana-to-QIE migration has four parts:

- **The channel structure maps across.** Iguana's From / Translator / To structure lines up with QIE's receiver, mapping and destination nodes, so channel scaffolding and connection configuration are converted rather than re-specified by hand.
- **The existing Lua is the specification.** The scripts already encode the transformations, routing rules and message handling the interface performs. They are read and interpreted, not discarded — which is what makes the requirements-gathering stage of a normal interface build largely unnecessary.
- **The AI Companion ports Lua to JavaScript.** The AI Companion is built into QIE and knows QIE. It writes JavaScript against QIE's own bindings, explains an existing script, answers development questions inside the console, and reads QIE's log output when something does not behave. Porting the transformation logic is the hard part of any Iguana migration, and it is where the Companion did the most work for Nexus.
- **A Qvera engineer validates before anything goes live.** The Companion assists your engineers rather than replacing them — a script it drafts enters a channel only when an engineer accepts it. Qvera reviews and validates the converted interfaces with you, so you go live with confidence, not surprises.

One thing does not come across. Connection credentials are encrypted in an Iguana export, so they are re-entered in QIE by hand on every migration and belong on the cutover checklist from the start.

How much of the existing work carried over?

All the interfaces Nexus ran in Iguana now run in QIE, supporting production workflows. The channel structure and connection configuration converted; the Lua transformation logic was ported to JavaScript with the AI Companion rather than rewritten from a blank editor, and a Qvera engineer validated each one before it went live.

The practical effect was that requirements gathering — normally the longest part of building an interface — was largely unnecessary, because the existing Lua already documented what each interface did.

What happened to message flow during the cutover?

Nothing stopped. Nexus moved the interfaces incrementally rather than in a single switchover, so at no point did the whole estate depend on one cutover working. The systems Nexus feeds saw no interruption in service.

For a health system that is the whole point: the platform underneath the data changed, and the data kept moving.

“

Our biggest concern going in was how much of our existing Iguana work would have to be redone. It was a much smaller lift than we thought. Our existing Lua code already defined what each interface needed to do, and the AI Companion handled the bulk of translating that into QIE scripts. Every interface is now running in production, with effectively no downtime.

Josh Atchley

Senior Manager, Platform & Infrastructure
Nexus Health Systems

What does this mean if you are running Iguana today?

Moving off Iguana does not have to mean discarding the work already invested in your integrations. Your Lua scripts, transformations, routing logic and interface requirements are the specification for the QIE build. The migration reads them rather than replacing them.

Every environment is different, and the only honest way to say what a move would take is to look at yours — how many channels, which transports, how much transformation logic sits in the Translator, and what has to cut over together.

Start with a free migration assessment

We review your current Iguana channels and interfaces, map them to QIE's channel model, plan the port of your Lua transformation logic, and give you a clear picture of effort, timeline and what you would gain. No obligation.

[Schedule a demo](#)[Request a free trial](#)

ABOUT QVERA

Qvera builds the Qvera Interface Engine (QIE) — a healthcare integration engine with a built-in AI Companion. QIE is in production today at more than 600 sites, running more than 25,000 interfaces and processing more than 50 million messages a day. It supports HL7 v2 and v3, FHIR, DICOM including native DIMSE, X12, NCPDP, ASTM, CDA and IHE profiles, and deploys on-premise, in containers, or in your own cloud. Qvera has been building healthcare integration software since 2008, is based in Kaysville, Utah, and is SOC 2 Type 2 attested.

ABOUT NEXUS HEALTH SYSTEMS

Since 1992, Nexus Health Systems has provided specialty care for patients with medical, behavioral and psychiatric needs. With a network of hospitals and medical-model residential treatment centers across Texas, Nexus serves children, teens and adults facing some of the most challenging diagnoses — including autism and related neurodevelopmental disorders, Prader-Willi Syndrome, neurological injuries and illnesses, and other medically complex conditions. Its interdisciplinary teams deliver fully integrated, whole-person care.

Produced by Qvera | www.qvera.com | © 2026

* HL7 is the registered trademark of Health Level Seven International. Iguana and INTERFACWARE are trademarks of their respective owners and are used here for identification only.