VertiGIS Studio and SOC 2 Type 2 Security Compliance

Updated July 20th, 2026

This compliance statement was prepared by Cam Barnard, Product Director for VertiGIS Studio.

Compliance Statement

VertiGIS North America Ltd. successfully completed its annual SOC 2 Type 2 audit in July 2026.

What is SOC 2 Type 2

SOC (System and Operational Controls) is an industry-recognized standard created by AICPA. SOC 2 deals specifically with the required cybersecurity controls necessary for secure, safe SaaS offerings. Type 2 means that controls and policies exist (type 1), and that an annual audit verifies that those controls and policies are being followed (type 2). An example of a system control is the regular scanning of code and libraries against the latest known security vulnerabilities. An example of an operational control is ensuring that all code is peer reviewed. System and operational controls work together to create a secure offering.

Benefits of SOC 2 Type 2 and VertiGIS Best Practices

The annual SOC 2 Type 2 audit and certification of the VertiGIS Studio SaaS environment (apps.vertigisstudio.com and apps.vertigisstudio.eu) combined with VertiGIS best practices provide value to all VertiGIS Studio customers.

The following table is not exhaustive, but it provides a good overview of how all VertiGIS Studio customers benefit.

	VertiGIS Studio SaaS	VertiGIS Studio customer-deployed
System and Operational Controls		
Third party network and application penetration testing including encryption, strong authentication, logging and other industry practices	●	
IAC, no manual adjustments to production environments	●	
Firewalls and intrusion detection and prevention systems	●	
Production environment vulnerability scans and patches	●	
SDLC all production code is peer reviewed	●	●
SDLC production code cannot be released by author	●	●
Regular vulnerability scans of code and libraries	●	●
Regularly reviewed roles and access permissions	●	●
Security incident response procedures	●	●
Anti-virus and anti-malware detection on all systems	●	●
Staff training for security awareness	●	●
VertiGIS Best Practices		
Static code scanning for known security vulnerabilities and code syntax patterns that may be problematic.	●	●
Unit tests run on every build. Confirms that function-level behavior and contracts have not changed.	●	●
Automated functional tests using the software as an end user does. Run weekly and at release. Rapid assessment of overall build quality, failed tests are re-run, investigated and resolved.	●	●
Pull Request (PR) Testing - PRs are peer-reviewed for errors and undergo thorough QA testing before a PR is approved and added to the master code base.	●	●
Human Regression Testing. At release, functional areas of the software are prioritized based upon change since last release. QA manually tests behavior starting with high priority and working down to low. Stakeholder bugs (reported by customers) are re-tested for the past several releases to ensure behavior remains correct.	●	●