Locate the translation layer in a Velocity network

A network can translate client traffic at a proxy or at supported backend installations. Adding another copy everywhere makes the topology harder to understand and does not guarantee broader support. First establish which component sees the original client and which version each backend actually runs.

Record the starting point

Draw the public proxy, lobby and destination servers on paper. Record their software builds and the Via plugins installed on each. Keep forwarding credentials out of this diagram. Include a working backend as a comparison, because the same client can expose different compatibility requirements across destinations.

Work through the change

1. Check startup logs to identify active ViaVersion installations and their platform. A JAR sitting in a folder is not proof that the plugin loaded successfully.
2. Compare the arrangement with the current ViaVersion platform guidance. For Velocity, inspect backend version detection in the generated configuration rather than assuming every destination uses the lobby version.
3. Join through the normal public proxy with the affected client and transfer to each backend. Record which transfer fails and the corresponding proxy and backend timestamps.
4. Trial a corrected supported placement on a private copy or controlled maintenance window. Change one translation location at a time and preserve protected backend forwarding.

Check the result

The chosen client should reach every destination you claim to support, and logs should identify one understood translation path for each hop. A successful lobby login alone is insufficient evidence. Include a native-version client in the test so the change does not hide a basic forwarding problem.

If the result differs

A failure on one destination suggests its version detection, plugin requirements or unsupported protocol direction needs checking. Authentication or forwarding errors are a separate issue. Installing ViaBackwards cannot repair a proxy secret mismatch or a backend listening on the wrong address.

Return to the recorded placement if the trial creates new failures. Leave direct backend access protected and retain both sides of the transfer log. Do not reset the network, player databases or login mode merely to simplify a compatibility diagram.

References and related help

- ViaVersion maintainer reference 1
- ViaVersion maintainer reference 2
- Allow compatible Minecraft Java clients with ViaVersion
- Set up a Velocity network with protected backend servers

Revision #1
Created 2026-11-09 16:36:26 UTC by CloudExa Hosting Admin
Updated 2026-11-09 16:36:26 UTC by CloudExa Hosting Admin