Build Private + Secure p2p Applications with Gosling

morgan (they/them)
morgan@blueprintforfreespeech.net

Gosling

- inspired by the RicochetIM lineage of instant messaging peer authentication

- onion-service to onion-service connectivity protocol and Rust library
  - solves the problem of verifying a connecting client also controls a particular onion-service

- build peer-to-peer applications+services with the following properties:
  - peer anonymity
  - end-to-end encryption
  - metadata resistance
  - NAT punching
  - censorship resistance
  - customisable additional peer authorisation

Updates!

Language Bindings

- C/C++ via the cgosling crate
- Java JNI bindings
- Python bindings

Example Chat Applications (Rust and C++)

```
Welcome to example_chat_rs!
Type help for a list of commands
> help
Available commands:
  help COMMAND              Print help for COMMAND
  init-context              Initialise the gosling context
  start-identity            Start the identity onion-service
  stop-identity             Stop the identity onion-service
  request-endpoint          Connect to identity onion-service and request an endpoint
  start-endpoint            Start an endpoint onion-service
  stop-endpoint             Stop an endpoint onion-service
  connect-endpoint          Connect to a peer's endpoint onion-service
  drop-peer                 Drop a connection to a peer
  list-peers                List all of the currently connected peers
  chat                      Send a message to a connected peer
  exit                      Quits the program
================================================================================
> init-context
```

Documentation

Design Document: **gosling.technology/design-doc.xhtml**
Usage Guide: **gosling.technology/usage-guide.xhtml**
Example: **github.com/blueprint-freespeech/gosling/tree/main/source/examples**
C/C++ API Reference: **gosling.technology/bindings/cgosling**
Java API Reference: **gosling.technology/bindings/goslingjni**

Additional Tor Configuration

- system tor
- bundled tor with additional configuration:
  - proxies
  - port allow-list
  - pluggable-transport + bridges
- in-process **arti-client** (experimental)

All usable on their own with the tor-interface crate: **crates.io/crates/tor-interface**

Tests and CI

- Rust crates fuzz-tested
- unit and functional tests for Rust crates, C/C++ library, and Java bindings
- tests running in GitHub CI on Windows, Linux, and macOS

Another Security Review

2024-11-12: **gosling.technology/security-reviews.xhtml**

What's Next?

- Further minor API improvements
- Bug Fixes
- Integration into Ricochet-Refresh

Try it!

- crates.io: **crates.io/crates/gosling**
- GitHub: **github.com/blueprint-freespeech/gosling**
- website: **gosling.technology**
  - specifications
  - guides
  - API documentation
  - examples