Department of Computer Science

BSc (Hons) Computer Science

Decentralized Peer-to-Peer SMPC Sealed-Bid Auction Marketplace over the Veilid Framework

Anker Rasmussen

anker.rasmussen@city.ac.uk

Project Consultant: Martin Nyx Brain

Client: Martin Nyx Brain (Academic Client)

Category: Academic Client Project

Subcategory: Application Development

Proprietary Interests: None

6 February 2026

Abstract

[To be written last. Summarise the problem (lack of privacy in online auctions), the approach (Veilid P2P framework + MP-SPDZ multiparty computation + sealed-bid protocol), key results (working prototype, N-party auctions tested, security analysis of side channels), and main conclusions.]

List of Figures

List of Tables

Glossary

AES-GCM	Advanced Encryption Standard – Galois/Counter Mode. Authenticated encryption scheme.
CBOR	Concise Binary Object Representation. A binary data serialization format.
DHT	Distributed Hash Table. A decentralized key-value store.
HE	Homomorphic Encryption. Encryption supporting computation on ciphertexts.
MPC / SMPC	(Secure) Multi-Party Computation. Protocols enabling joint computation over private inputs without revealing them.
MP-SPDZ	Multi-Protocol SPDZ. A framework for multi-party computation supporting multiple protocol backends.
NAT	Network Address Translation.
P2P	Peer-to-Peer.
SPDZ	"Speedz" protocol for MPC with dishonest majority security.
SSS	Shamir's Secret Sharing. A threshold secret-sharing scheme.
Veilid	A privacy-first, open-source P2P application framework with onion routing and a secure DHT.

Contents

1 Introduction

1.1 Problem Statement
1.2 Project Objectives
1.3 Beneficiaries
1.4 Scope and Assumptions
1.5 Report Structure

2 Output Summary

3 Context

3.1 Online Auction Mechanisms

3.2 Secure Multi-Party Computation

3.3 Privacy-Preserving Peer-to-Peer Networks

3.4 Commitment Schemes and Content Encryption

4 Method

4.1 Development Approach

4.2 System Architecture

4.3 Technology Choices and Justifications

4.3.1 Veilid for Coordination and Transport

4.3.2 Shamir Secret Sharing for MPC

4.3.3 Commitment Scheme for Bid Integrity

4.3.4 Serialization Formats

4.3.5 Desktop UI Framework

4.4 MPC Protocol Design

4.5 Testing Strategy

4.6 Reused Software

5 Results

5.1 System Architecture

5.2 Auction Protocol

5.3 MPC Integration

5.4 User Interface

5.5 Testing and Verification

5.6 Security Analysis

5.6.1 Threat Model

5.6.2 Timing Side Channel on Party Assignment

5.6.3 Seller as Single Point of Trust

5.6.4 Winner Identity Linkage

5.6.5 DHT Metadata Leakage

5.6.6 MPC Route Correlation

5.6.7 Small-N Threshold Vulnerability

5.6.8 Summary of Side Channels

6 Conclusions and Discussion

6.1 Objectives Revisited
6.2 Discussion
6.3 Legal, Social, Ethical and Professional Issues
6.4 Future Work
6.5 Reflections

Appendix A Project Definition Document

Project Definition Document

November 10, 2025

Project Title: Decentralized Peer-to-Peer SMPC Sealed-Bid Auction App over the Veilid framework

Degree Programme: BSc (Hons) Computer Science

Project Consultant: Martin Nyx Brain

By: Anker Rasmussen

anker.rasmussen@city.ac.uk

Category: Academic Client Project

Subcategory: Application Development

Word count: 1631

Proprietary Interests: None

Contents

1 Proposal

1.1 Problem to be Solved

Popular online consumer marketplaces either use *open, incrementally visible bidding* or *no auctions at all*, which undermines sealed-bid privacy and fairness. Current popular marketplaces such as eBay and Facebook Marketplace follow two distinct patterns. eBay's proxy-bidding model reveals live bid progression and awards the item to the highest bid at close [1]. Facebook Marketplace, by contrast, is a listing-plus-messaging workflow rather than a formal auction protocol, with weak buyer protections [2]. Public blockchain auctions add strong auditability but invite transaction ordering attacks (front-running/MEV) that can leak or distort bids during submission and reveal phases [3]. Cryptographic sealed-bid auctions can hide non-winning bids entirely. Deploying a protocol without a trusted auctioneer requires secure multiparty computation (MPC). MPC has been demonstrated in production (e.g., the Danish sugar-beet auction) and modern frameworks such as MP-SPDZ show practical performance across secret-sharing, HE, and garbled-circuit backends [4, 5, 6, 7, 8].

What is Veilid?

Veilid is a privacy-first, open-source, peer-to-peer application framework. Each app embeds a node into a global overlay where peers are equal (no privileged relays), connections are end-to-end encrypted, and private routing obscures network locations. After a brief bootstrap, apps communicate directly over transports such as UDP/TCP/QUIC/Web and exchange data via a secure DHT designed for mobile and desktop [9]. Veilid provides addressable, encrypted endpoints (public-key identities) and metadata-minimized communication, independent of any blockchain.

Why Veilid for this marketplace?

In line with the project brief to *build an application using the Veilid framework*, this marketplace needs private identities, censorship-resistant transport, and NAT-friendly reachability without a trusted coordinator. Veilid provides exactly these: (i) addressable, encrypted endpoints keyed by public keys rather than IPs, (ii) a secure DHT for publishing listings and discovering MPC parties, and (iii) obfuscated routing that limits metadata exposure during bid submission and MPC setup [9]. Because Veilid is blockchain-agnostic, settlement can occur off-overlay (e.g., Monero testnet) while *coordination* and *communication*

remain private and decentralized. [10].

Project Goal

Build a **peer-to-peer sealed-bid marketplace** that (1) preserves bidder privacy by default, (2) mitigates front-running risk during bid submission, and (3) operates *without a central auctioneer*. The prototype combines MPC for winner/price computation with *Veilid* for identity, routing, and availability [9]. Payments clear on *Monero testnet* after the MPC outcome is published. This demonstrates the full flow without handling real funds while inheriting the confidential-transaction model from the CryptoNote design (Monero precursor) [10].

1.2 Project Objectives

- **Main objective.** This project shall deliver a working peer-to-peer marketplace application on the Veilid network that exclusively supports sealed-bid listings and private content unlock for the winning bidder.
- **Testable sub-objectives.**
  - *Listings and purchases.*
    Implement end-to-end flows: create listing → submit bid → determine winner → complete purchase.
    Test: demo run and automated integration tests covering success/edge cases (invalid bid, tie, timeout).
  - *Sealed-bid via MPC.*
    Integrate a multi-party computation protocol to select the highest valid bid without revealing non-winning bids.
    Test: unit tests with mocked parties. Tests to show loser-bid privacy with reproducible benchmarks for N bidders.
  - *MPC-gated decryption.*
    Ideally: Bind winner selection to content access: the MPC that selects the highest valid bid also releases a winner-only decryption capability for the listing (K via 1-out-of-n OT(Oblivious Transfer)) only if settlement is confirmed on testnet.
    Test: unit tests showing that non-winners cannot recover K. Create integration test with mocked `paid`=0/1. Testnet demo where confirmation flips release.
  - *Settlement on Monero Testnet.*
    Simulate network init, fund wallets, and programmatic "real" transactions for deposits/escrow/release.

Test: scripted testnet transactions with confirmations & failure handling.

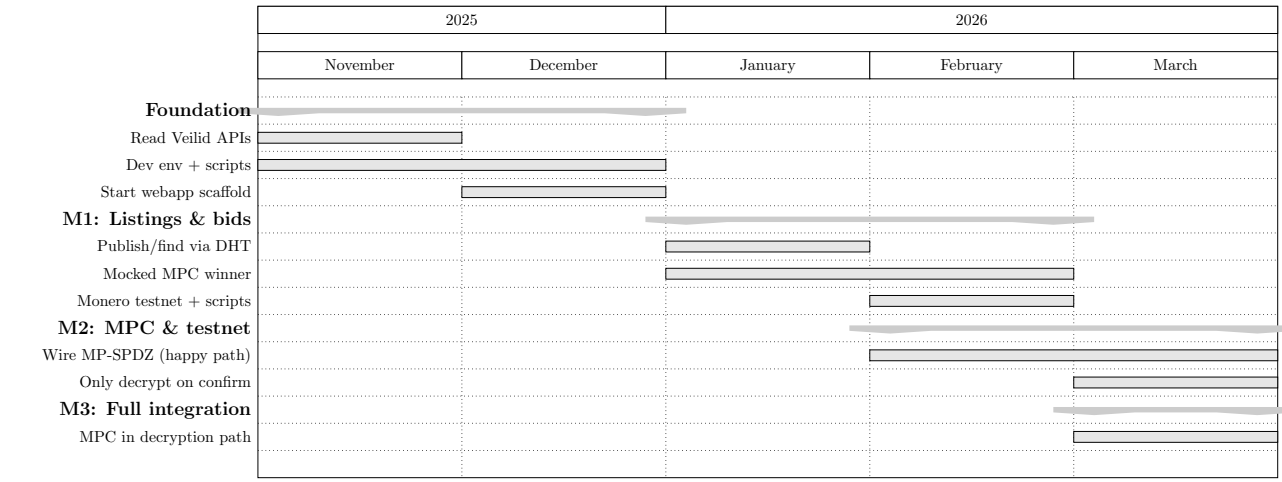
1.3 Project Beneficiaries

- **Project supervisor:** Gain a proof-of-concept application that utilizes Veilid's framework (keyed identities, secure DHT, obfuscated routing) with a non-toy use case. Deliverables (repo and demo video, alongside shell scripts) support future supervision of similar projects.
- **Veilid development team/Veilid Community:** A real world application that demonstrates capabilities of Veilid when integrated with other academic projects. Potential upstream bugfixes if warranted.
- **Marketplace users - research oriented (prototype).** Sellers get fairer price discovery without leaking losing bids. Bidders gain privacy and reduced front-running risk. In real-world deployments, at-risk users (e.g., in high-censorship environments) could benefit from metadata-minimising coordination.
- **End users in the future.** A metadata-minimising, sealed-bid coordination layer could be adapted for *safer* exchange of sensitive information (e.g., purchasing or rewarding data disclosures) where overt communication is filtered or surveilled.

1.4 Project Plan

- **Foundation (Nov-Dec 2025): Understand & set up Veilid. Digest current literature**
  - Read Veilid API spec & pull core code segments from *veilidchat*.
  - Create and ready a platform agnostic development environment with scripts to boot local Veilid nodes (to aid my cross-platform development - x86 Linux/MacOS ARM).
  - Begin development of webapp, light tests to see dev env works before implementing any of the features to be delivered. Stack: C++ to WebAssembly with a TypeScript UI.
- **Milestone 1 (mid Jan–Feb 2026): Listings & bids over Veilid**
  - Listing publish/discovery over secure DHT & private bid submission path.
  - Mocked MPC winner & basic content encryption at listing. Create happy-path E2E test.
  - Spool up private Monero testnet and understand integration with Veilid.
  - Update Monero testnet script & modify to fit usecase. Integrate with deployment script of dev network.

- **Milestone 2 (late Feb–Mar 2026): MPC & testnet**
  - Integrate real MPC (MP-SPDZ) and test happy path.
  - Fully integrate Monero testnet scripts. Implement Winner-only decrypt on confirmation.
  - E2E demo with simple implementation of MP-SPDZ + testnet.
- **Milestone 3 (late Mar 2026): Fully integrated MPC in Decryption**
  - (Attempt to) fully integrate MPC to the decryption algorithm.
  - Make test scripts/infrastructure spool up for project demo (dockerized nodes on one device).
- **Deliverables (Apr 2026): Package & submit**
  - Wrap up writing dissertation
  - Submit final report & demo video.


2025
2026
November
December
January
February
March
Foundation
Read Veilid APIs
Dev env + scripts
Start webapp scaffold
M1: Listings & bids
Publish/find via DHT
Mocked MPC winner
Monero testnet + scripts
M2: MPC & testnet
Wire MP-SPDZ (happy path)
Only decrypt on confirm
M3: Full integration
MPC in decryption path

1.5 Risk Matrix (Probability–Severity–Mitigation)

Legend: L = Low, M = Medium, H = High.

Risk	Prob.	Sev.	Mitigation
Veilid API or reusing code from `veilidchat` is harder than expected	M	H	Build a small wrapper with four actions: publish, find, send, receive. Start by running `veilidchat` as a separate helper and talk to it with simple text messages. If the direct API later works, replace the helper. Write clear build steps and ensure code is commented and documented to reduce code entropy (loss of context).
Understanding MPC and integrating MP–SPDZ takes longer than planned	M	H	Follow a short learning plan: run the MP–SPDZ tutorial and a simple max-of-N example. Treat MPC as a small command-line tool with JSON in/out so the app can call it easily. Run all parties locally first, before implementing dockerised build scripts.
Integrating Veilid–MPC–Monero (HTTP/JSON)	M	H	Standardise on HTTP+JSON: use Monero's native JSON-RPC, expose MPC as a localhost service with a minimal OpenAPI/JSON-Schema-defined surface. Generate clients, add contract tests before E2E, enforce timeouts/retries and idempotent ops. Keep MPC/Monero as separate processes.
Containerisation & multi-arch builds (Docker/Compose)	M	H	Build small images from fixed base versions. Use plain HTTP/JSON between services so they connect cleanly. Start everything with a single Compose file (Veilid app, MPC sidecar, Monero sidecar). Keep chain data in named volumes. To reduce surprises, target one platform for the demo (64-bit Linux) and have CI build and run the images throughout development. Add health checks and restart policies, and run as a non-root user with secrets loaded from a .env file.

Continued on next page

Risk	Prob.	Sev.	Mitigation
P2P connectivity on eduroam	M	M	Dockerise apps on same-LAN with docker multi-node compose & keep a pre-recorded run with logs & video for assessment.
Toolchain/portability (macOS vs Linux, dependencies)	M	M	Containerised builds with fixed minimal versions. CI on x86_64 Linux on GitHub Actions, avoiding platform-specific flags. reproducible build script for different architectures.
Hardware failure / data loss	L	H	Keep dissertation sourcefiles in `.tex` within Git, push & pull frequently. Keep remote backups for code in GitHub & GitLab. Maximum loss will be minimal

1.6 Legal, Social, Ethical and Professional Considerations

Project category (per handbook 4.5.7): Category 1 prototype not intended for distribution. Operation is confined to a local Veilid dev overlay and Monero testnet. No mainnet, no PII, and no real funds.

Scope. Proof-of-concept only: runs on testnet for Monero, but runs on a localized Veilid dev environment.

Statement on illicit uses. I explicitly condemn any attempt to adapt this research for illicit trade, exploitation, or other illegal activity. The work is motivated by technical research questions (privacy-preserving auction protocols, MPC integration, and overlay networking) rather than any intent to facilitate wrongdoing. All design, testing and evaluation are performed within controlled, local test environments to prevent misuse.

Principal risk (dual-use). A private marketplace design could in principle be repurposed to facilitate illicit exchange if deployed without appropriate safeguards.

Mitigations to be implemented in this prototype.

- No mainnet support. Script will have localized test peers (on same LAN) with public bootstrap for Veilid disabled (dev veilid network).
- No custody of real funds, testnet. Payment "confirmations" exist only on Monero's testnet, and wallets/the network are created/destroyed with the run script.

Legal (licensing). Project is released under **MPL-2.0** (file-level copyleft). Veilid

already exists under **MPL-2.0**. any modified Veilid files are published under MPL as required. No copyleft extends to independent project files beyond their chosen license.

Professional conduct. No human participants.

Residual risk and fallback. Code could be forked and modified. Reduce this by omitting deployment scripts on live Veilid + live Monero. If further legal/ethical concerns arise, disable network capabilities and run on a local area network.

1.7 References

References

[1] eBay Help, *How bidding works*, Accessed 29 Oct 2025, 2025. [Online]. Available: https://www.ebay.co.uk/help/buying/bidding/bidding?id=4003.

[2] Meta/Facebook Help Center, *Marketplace help center: Using marketplace to buy and sell*, Accessed 29 Oct 2025, 2025. [Online]. Available: https://www.facebook.com/help/1713241952104830.

[3] P. Daian, S. Goldfeder, T. Kell, Y. Li, X. Zhao, I. Bentov, L. Breidenbach, and A. Juels, *Flash boys 2.0: Frontrunning, transaction reordering, and consensus instability in decentralized exchanges*, 2019. DOI: 10.1109/SP40000.2020.00040. arXiv: 1904.05234 [cs.CR]. [Online]. Available: https://arxiv.org/abs/1904.05234.

[4] P. Bogetoft, D. L. Christensen, I. Damgård, M. Geisler, T. Jakobsen, M. Krøigaard, J. D. Nielsen, J. B. Nielsen, K. Nielsen, J. Pagter, M. Schwartzbach, and T. Toft, "Secure multiparty computation goes live," in *Financial Cryptography and Data Security*, R. Dingledine and P. Golle, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 325–343, ISBN: 978-3-642-03549-4. DOI: 10.1007/978-3-642-03549-4_20. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-642-03549-4_20.

[5] M. Keller, "Mp-spdz: A versatile framework for multi-party computation," in *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '20, Virtual Event, USA: Association for Computing Machinery, 2020, pp. 1575–1590, ISBN: 9781450370899. DOI: 10.1145/3372297.3417872. [Online]. Available: https://doi.org/10.1145/3372297.3417872.

[6] I. Damgård, V. Pastro, N. Smart, and S. Zakarias, "Multiparty computation from somewhat homomorphic encryption," in *Advances in Cryptology – CRYPTO 2012*, R. Safavi-Naini and R. Canetti, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 643–662, ISBN: 978-3-642-32009-5.

[7] D. Evans, V. Kolesnikov, and M. Rosulek, "A pragmatic introduction to secure multi-party computation," *Found. Trends Priv. Secur.*, vol. 2, no. 2–3, pp. 70–246, Dec. 2018, ISSN: 2474-1558. DOI: 10.1561/3300000019. [Online]. Available: https://doi.org/10.1561/3300000019.

[8] K. Sako, "An auction protocol which hides bids of losers," in *Public Key Cryptography*, H. Imai and Y. Zheng, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2000, pp. 422–432, ISBN: 978-3-540-46588-1. DOI: 10.1007/978-3-540-46588-1_28. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-540-46588-1_28.

[9] Veilid Project. "Veilid developer book." Official developer documentation for the Veilid framework, Accessed: Oct. 31, 2025. [Online]. Available: https://veilid.gitlab.io/developer-book/.

[10] N. van Saberhagen, "Cryptonote v 2.0," Original CryptoNote whitepaper; commonly cited in Monero literature, Oct. 17, 2013. Accessed: Oct. 31, 2025. [Online]. Available: https://decred.org/research/saberhagen2013.pdf.

1.8 Appendix: Research Ethics Form

Research Ethics Review Form for BSc and MSci Projects

Computer Science Research Ethics Committee (CSREC)

http://www.city.ac.uk/department-computer-science/research-ethics

Undergraduate students undertaking their final project in the Department of Computer Science must consider the ethics of their project work and ensure that it complies with research ethics guidelines and the law for data protection. In some cases, a project will need approval from an ethics committee before it can proceed. Usually, but not always, this will be because the student is involving other people ("participants") in the project.

To ensure that they give appropriate consideration to ethical issues, all students must complete this form and attach it to their project definition document (PDD). There are two parts:

PART A: Ethics Checklist. All students must complete this part.
The checklist identifies whether the project requires ethical approval and, if so, where to apply for approval.

PART B: Ethics Proportionate Review Form. Students who have answered "no" to all questions in A1, A2 and A3 and "yes" to question 4 in A4 in the ethics checklist must complete part B as well. The project supervisor or consultant has delegated authority to provide approval in such cases that are considered to involve MINIMAL risk. The approval may be ***provisional*** – *identifying the planned work with human end user participants as* likely to involve MINIMAL RISK. In such cases you must additionally seek ***full approval*** from the supervisor or consultant as the project progresses and details are established. You must obtain ***full approval*** in writing, before recruiting and engaging with human end users participants for your project.

A.1 If you answer YES to any of the questions in this block, your consultant/supervisor must have obtained approval for the project from an appropriate external ethics committee, and you need to have received written confirmation of this from him/her. Students cannot themselves apply for ethics approval in this case as the project is considered high risk". This type of research is not covered by City's process, and external approval from an appropriate institution is required.		*Delete as appropriate*
1.1	Does your research require approval from the National Research Ethics Service (NRES)?	**NO**
1.2	Will you recruit participants who are covered by the Mental Capacity Act 2005?	**NO**
1.3	Will you recruit any participants who are covered by the Criminal Justice System, for example, people on remand, prisoners and those on probation?	**NO**
A.2 If you answer YES to any of the questions in this block your consultant/supervisor must have obtained appropriate ethics committee approval		*Delete as appropriate*
2.1	Does your research involve participants who are unable to give informed consent? *For example, people who may have a degree of learning disability or mental health problem, that means they are unable to make an informed decision on their own behalf.*	**NO**
2.2	Is there a risk that your research might lead to disclosures from participants concerning their involvement in illegal activities?	**NO**
2.3	Is there a risk that obscene and or illegal material may need to be accessed for your research study (including online content and other material)?	**NO**
2.4	Does your project involve participants disclosing information about protected characteristics (as identified by the Equality Act 2010)?	**NO**

	For example: racial or ethnic origin; political opinions; religious beliefs; trade union membership; physical or mental health; sexual life; criminal offences and proceedings	
2.5	Does your research involve you travelling to another country outside of the UK, where the Foreign & Commonwealth Office has issued a travel warning that affects the area in which you will study? *Please check the latest guidance from the FCO - http://www.fco.gov.uk/en/*	**NO**
2.6	Does your research involve invasive or intrusive procedures? *These may include, but are not limited to, electrical stimulation, heat, cold or bruising.*	**NO**
2.7	Does your research involve animals?	**NO**
2.8	Does your research involve the administration of drugs, placebos or other substances to study participants?	**NO**
A.3 If you answer YES to any of the questions in this block, then unless you are applying to an external ethics committee or the Senate Research Ethics Committee (SREC), you must apply for approval from the Computer Science Research Ethics Committee (CSREC) through Research Ethics Online - https://researchmanager.city.ac.uk/. Depending on the level of risk associated with your application, it may be referred to the Senate Research Ethics Committee (SREC).		*Delete as appropriate*
3.1	Does your research involve participants who are under the age of 18?	**NO**
3.2	Does your research involve adults who are vulnerable because of their social, psychological or medical circumstances (vulnerable adults)? *This includes adults with cognitive and / or learning disabilities, adults with physical disabilities and older people.*	**NO**
3.3	Are participants recruited because they are staff or students of City, University of London? *For example, students studying on a particular course or module.* *If yes, then approval is also required from the Head of Department or Programme Director.*	**NO**
3.4	Does your research involve intentional deception of participants?	**NO**
3.5	Does your research involve participants taking part without their informed consent?	**NO**
3.5	Is the risk posed to participants greater than that in normal working life?	**NO**
3.7	Is the risk posed to you, the researcher(s), greater than that in normal working life?	**NO**
A.4 If you answer YES to the following question and your answers to all other questions in sections A1, A2 and A3 are NO, then your project is deemed to be of MINIMAL RISK. **If this is the case, then you can apply for approval through your supervisor under PROPORTIONATE REVIEW. You do so by completing PART B of this form.** **If you have answered NO to all questions on this form, then your project does not require ethical approval. You should submit and retain this form as evidence of this.**		*Delete as appropriate*
4	Does your project involve human participants or their identifiable personal data? *For example, as interviewees, respondents to a survey or participants in testing.*	**NO**

1.9 Appendix: Client Description Sheet

“‘ This project involves working with an academic at City St George’s in the department of computer science. The academic will supervise and guide your work alongside being your consultant and marker.

Contact details:

Staff name:	Martin Nyx Brain
Email address:	martin.brain@citystgeorges.ac.uk

Project title: Application Development Using the Veilid Application Framework

Project description:	Keywords: networking, privacy, anonymity, peer-to-peer, distributed applications, anti-surveillance, cryptography
	Description: The Veilid framework (https://veilid.com/) is a recent developed collection of protocols to build distributed, private applications. It builds on the onion routing ideas used by Tor and I2P as well as distributed hash tables and routing as used in IPFS and others. It aims to make developing highly-scalable, distributed, private applications easy. Veilid is still very much under active development, so this project would aim to create a proof-of-concept / example application to demonstrate some of its features. Possible applications:
	• A multi-user notepad / text editor.
	• An anonymous voting system.
	• An anonymous question submission system.
	• A SOCKS proxy
	Plan:
	1. Learn the key concepts involved including onion routing and distributed hash tables.
	`2. Set up a Veilid node and try out the demo application.`
	
	`3. Develop your own application!`
	Skills: Knowledge of networking and cryptography, software development on Linux, reasonable programming ability.

Ideal Project Outcomes:	1. A network application that uses the Veilid Framework. 2. A demonstration of the application.
Additional Information	Contact Martin for more information.

Appendix B Reuse Summary

This appendix documents all third-party software reused in this project, as required by the handbook.

Frameworks and Libraries

Component	Version	Licence	Usage
veilid-core	(local build)	MPL-2.0	P2P networking, DHT, private routing
MP-SPDZ	(local build)	BSD-3-Clause	Shamir MPC protocol execution
Dioxus	0.7.2	MIT/Apache-2.0	Desktop UI framework
tokio	1.x	MIT	Async runtime
serde	1.0	MIT/Apache-2.0	Serialization framework
ciborium	0.2	MIT/Apache-2.0	CBOR serialization for DHT records
bincode	1.3	MIT	Binary serialization for messages
sha2	0.10	MIT/Apache-2.0	SHA-256 hashing (bid commitments)
aes-gcm	0.10	MIT/Apache-2.0	AES-256-GCM content encryption
rand	0.8	MIT/Apache-2.0	Random number generation
anyhow	1.0	MIT/Apache-2.0	Error handling
thiserror	1.0	MIT/Apache-2.0	Error type derivation
tracing	0.1	MIT	Structured logging
async-trait	0.1	MIT/Apache-2.0	Async trait support

Table B.1: Third-party libraries used in the market crate.

Development Tools

- **Docker / Docker Compose**: Veilid devnet environment (5-node topology).
- **cargo-nextest**: Test runner for parallel and sequential test execution.
- **libipspoof.so**: LD_PRELOAD library from Veilid devcontainer for IP address spoofing in devnet. Source: Veilid repository (MPL-2.0).

Code Reuse from Reference Applications

The Veilid project's `veilidchat` application was studied as a reference for Veilid API usage patterns (DHT record creation, routing context setup, message handling). No code was directly copied; API patterns were adapted for the auction use case.

Appendix C Supervisory Meeting Minutes

Meeting 1 — [Date]

Attendees: Anker Rasmussen, Martin Nyx Brain

Discussion: [Discussion about decryption, how to handle decryption communication over nodes, AppBroadcast vs AppMessage (veilid implementation)]

Actions: [Discussion about]

Meeting 1 — [Date]

Attendees: Anker Rasmussen, Martin Nyx Brain

Discussion: [Leaky secret transfer, metadata leakage.]

Actions: [Agreed actions]

Meeting 1 — [Date]

Attendees: Anker Rasmussen, Martin Nyx Brain

Discussion: [Summary]

Actions: [Agreed actions]