Patient Explorer Application

Security & Architecture Overview

Prepared for: Pat (IT Review) & Brian (Technical Review) **Prepared by**: Robert Green, MD **Date**: December 8, 2025 **Version**: 1.0

Executive Summary

Patient Explorer is a **HIPAA-compliant patient consent tracking and data migration tool** developed for Green Clinic to manage patient records during EMR transitions. The system is designed with multiple layers of security to protect Protected Health Information (PHI) while enabling efficient workflows across multiple workstations.

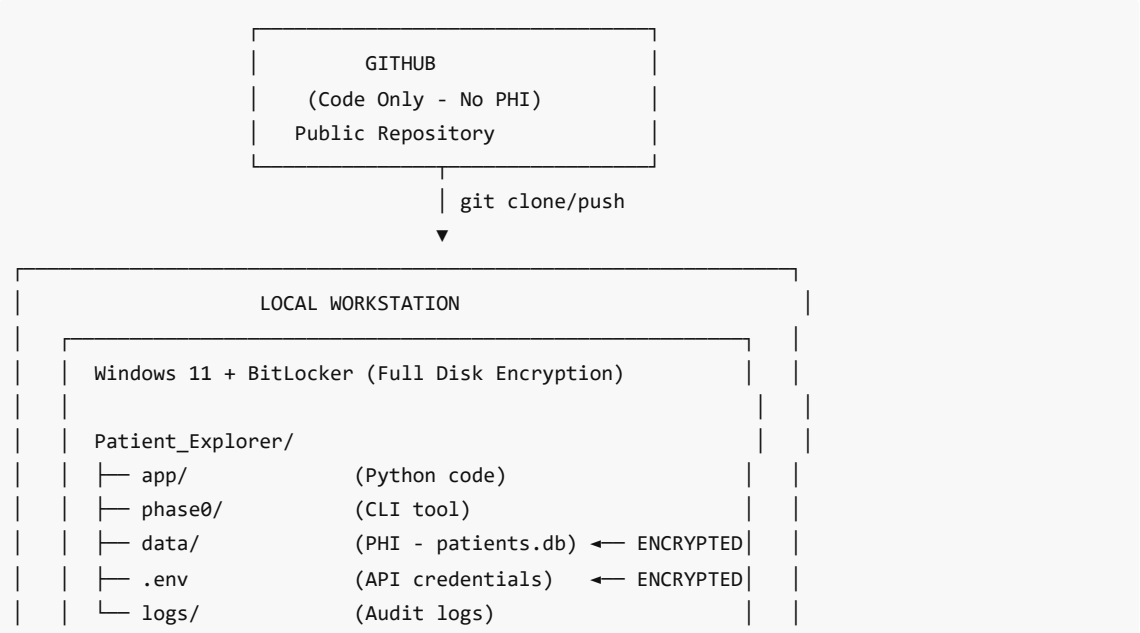
Key Security Highlights

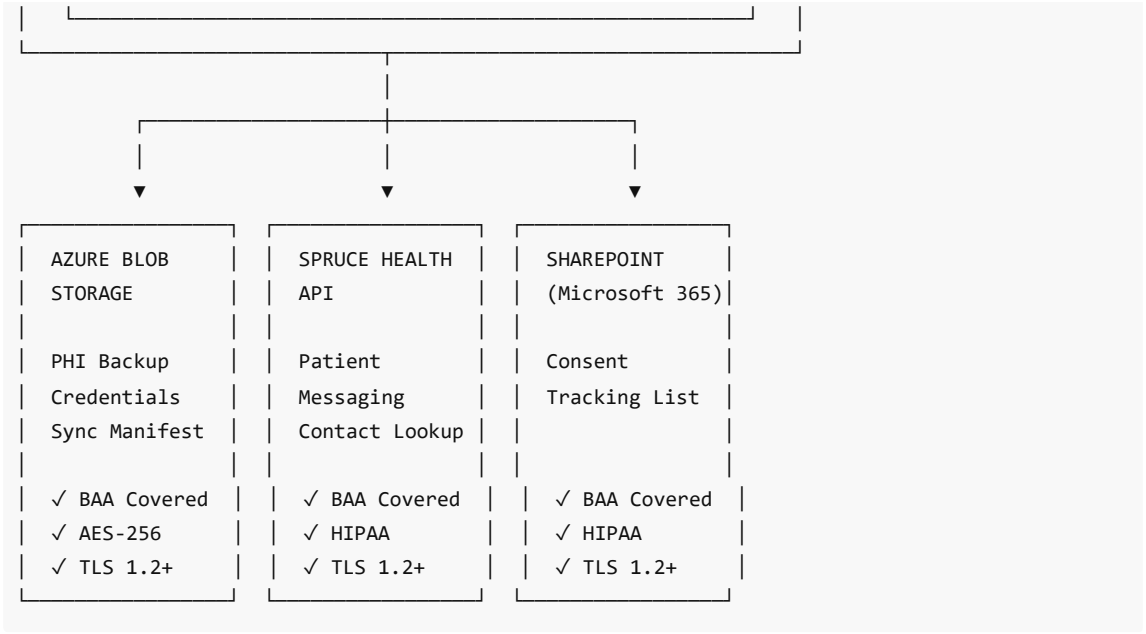
Feature	Implementation
Data Encryption	AES-256 at rest, TLS 1.2+ in transit
Local Device Security	Windows 11 + BitLocker full-disk encryption
Cloud Storage	Azure Blob Storage with HIPAA-compliant configuration
Authentication	Azure Active Directory (AD) with RBAC
BAA Coverage	Microsoft 365, Azure, and Spruce Health under signed BAAs
Code Repository	GitHub (no PHI stored in code)

---

1. System Architecture

Data Flow Diagram





What Goes Where

Data Type	Storage Location	Encryption	BAA
Source code	GitHub	N/A (no PHI)	N/A
Patient database	Local + Azure Blob	BitLocker + AES-256	Microsoft
API credentials	Local + Azure Blob	BitLocker + AES-256	Microsoft
Patient contacts	Spruce Health	TLS 1.2+	Spruce
Consent tracking	SharePoint	Microsoft 365 encryption	Microsoft

---

2. Security Controls

2.1 Local Device Security

Control	Requirement	Status
Operating System	Windows 11 Pro/Enterprise	Required
Disk Encryption	BitLocker enabled	Required
User Authentication	Windows Hello / PIN	Required
Auto-Lock	5 minute timeout	Configured

2.2 Azure Blob Storage Security

Storage Account: `stgreenclinicworkspace` **Region**: East US 2 (HIPAA-compliant data center) **Redundancy**: Geo-Redundant Storage (GRS) for disaster recovery

Security Layer	Configuration

Encryption at Rest	AES-256 (Microsoft-managed keys)
Encryption in Transit	TLS 1.2+ enforced (HTTPS only)
Authentication	Azure Active Directory
Authorization	Role-Based Access Control (RBAC)
Anonymous Access	Disabled
Soft Delete	7-day recovery window
Blob Versioning	Enabled (file history)
Public Access	Disabled at account level

2.3 Access Control

Authorized Users:

User	Role	Access Level
rgreen@greenclinicteam.com	Owner	Storage Blob Data Contributor
rgreen@southviewteam.com	Admin	Storage Blob Data Contributor

Adding New Users:

1. Azure Portal → Storage Account → Access Control (IAM)
2. Add role assignment: "Storage Blob Data Contributor"
3. Assign to user's Microsoft account email

2.4 Compliance Tagging

All Azure resources are tagged for compliance tracking:

Tag	Value
Environment	Production
Project	Patient_Explorer
Purpose	PHI-workspace-sync
Compliance	HIPAA
Owner	rgreen@greenclinicteam.com
CostCenter	Green_Clinic

---

3. Application Features

3.1 Current Capabilities (Phase 0)

Feature	Description	Security Notes

Patient List Import	Load patients from Excel	PHI stays local, only stats shown
Spruce Matching	Match patients to Spruce contacts	API call over TLS, results to local file
Consent Tracking	Track consent status in SharePoint	Microsoft 365 BAA coverage
Multi-Device Sync	Sync workspace between devices	Azure Blob with HIPAA config

3.2 CLI Commands

```
# Patient Management
python -m phase0 test-spruce          # Test API connection
python -m phase0 load-patients <file> # Import patient list
python -m phase0 match-spruce <file>  # Match to Spruce contacts
python -m phase0 status               # Show consent statistics

# Workspace Sync (Azure)
python -m phase0 sync-push --interactive   # Upload to Azure
python -m phase0 sync-pull --interactive   # Download from Azure
python -m phase0 sync-status --interactive # Check sync status
```

3.3 HIPAA-Compliant Output

All terminal output shows **aggregate statistics only**:

```
✓ GOOD: "Matched: 45 of 120 patients (37.5%)"
✗ BAD:  "Found: John Smith, 555-123-4567"
```

Detailed patient data is written to **local files only**, never displayed in terminal or sent to AI assistants.

---

4. Business Associate Agreements

Service Provider	BAA Status	Use Case
Microsoft Azure	Signed	PHI storage, sync, backup
Microsoft 365	Signed	SharePoint consent tracking
Spruce Health	Signed	Patient messaging, contact lookup
Anthropic (Claude)	NOT Signed	Code assistance only - NO PHI

Important: AI Assistant Restrictions

Claude Code (Anthropic) does **NOT** have a BAA. Therefore:

- Never paste patient data into Claude Code chat
- Never display PHI in terminal output
- All PHI processing happens in local Python scripts
- AI sees code and aggregate stats only

---

5. Multi-Device Sync Workflow

Switching to a New Device

```
# On current device (before leaving)
python -m phase0 sync-push --interactive

# On new device
git clone https://github.com/RGgreenbhm/Patient_Explorer.git
cd Patient_Explorer
python -m venv .venv
.venv\Scripts\activate
pip install -r requirements.txt
python -m phase0 sync-pull --interactive
```

What Gets Synced

Synced to Azure	NOT Synced (Rebuild Locally)
`data/patients.db`	`.venv/` (Python environment)
`.env` (credentials)	`__pycache__/` (cache)
`logs/*.log`	IDE settings
`.gitignore-sync.json`	

Authentication Flow

1. User runs sync command with `--interactive` flag
2. Browser opens to Microsoft login page
3. User authenticates with authorized Azure AD account
4. Token returned to application
5. Sync proceeds with authorized credentials

---

6. Security Verification Checklist

Azure Storage Account

- [x] Secure transfer (HTTPS) required
- [x] Minimum TLS version 1.2
- [x] Public blob access disabled
- [x] Azure AD authentication required
- [x] RBAC permissions configured
- [x] Soft delete enabled (7-day recovery)
- [x] Blob versioning enabled
- [x] Geo-redundant storage (GRS)
- [x] HIPAA compliance tags applied
- [x] Default OAuth authentication enabled

Local Workstation

- [x] Windows 11 Pro/Enterprise
- [x] BitLocker enabled
- [x] Auto-lock configured
- [x] Antivirus active
- [x] Firewall enabled

Application

- [x] No PHI in terminal output
- [x] No PHI in git repository
- [x] Credentials in .env (gitignored)
- [x] SHA256 hashing for change detection
- [x] Audit logging enabled

---

7. Incident Response

Data Breach Protocol

1. **Immediate**: Revoke Azure access tokens
2. **Within 1 hour**: Notify practice administrator
3. **Within 24 hours**: Document incident details
4. **Within 72 hours**: Notify affected patients (if required)

Recovery Procedures

1. **Lost Device**: Remote wipe via Microsoft Intune (if enrolled)
2. **Corrupted Data**: Restore from Azure Blob (soft delete / versioning)
3. **Credential Exposure**: Rotate all API keys, update .env, re-sync

---

8. Cost Summary

Resource	Monthly Cost
Azure Storage (Standard GRS, ~1GB)	~$0.50
Azure Operations (read/write)	~$0.05
Total Azure	**< $1/month**

Note: Microsoft 365 and Spruce Health costs are covered under existing subscriptions.

---

9. Future Enhancements

Planned

Enhancement	Purpose	Timeline
Azure Key Vault	Production credential management	Before production

Azure Storage Analytics	Enhanced audit logging	Q1 2026
Multi-factor authentication	Additional security layer	As needed

Production Migration Path

1. Create Azure Key Vault in `Green_Clinic` resource group
2. Store API tokens as Key Vault secrets
3. Update application to fetch secrets at runtime
4. Remove raw credentials from blob sync
5. Enable Key Vault access logging

---

10. Support Contacts

Role	Contact
Azure Admin	rgreen@greenclinicteam.com
Application Owner	Robert Green, MD
Microsoft Support	Azure Portal → Help + Support
Spruce Health Support	support@sprucehealth.com

Appendix A: Azure Resource Details

Property	Value
Subscription	PAYG-RG-Dev
Subscription ID	ec8ffbbb-516a-42e9-8489-9c2245954a0d
Resource Group	Green_Clinic
Storage Account	stgreenclinicworkspace
Container	workspace-sync
Region	East US 2
SKU	Standard_RAGRS
URL	https://stgreenclinicworkspace.blob.core.windows.net

Appendix B: Technology Stack

Component	Technology	Version
Runtime	Python	3.12+
CLI Framework	Typer + Rich	Latest
Data Models	Pydantic	2.x

Excel Import	pandas + openpyxl	Latest
HTTP Client	httpx	Latest
Azure SDK	azure-storage-blob, azure-identity	12.x, 1.x
SharePoint	Office365-REST-Python-Client	2.x
Logging	loguru	Latest

Document Generated: December 8, 2025 Classification: Internal Use - Contains Security Configuration Details Review Schedule: Quarterly or after significant changes