

SERVICE 5

Custom Framework Development & Automation Services

Service Definition Document

G-Cloud15 Procurement Submission

1. SERVICE OVERVIEW

TestFyra develops custom frameworks, automation platforms, and bespoke tools tailored to your specific operational needs. Unlike off-the-shelf software, our custom frameworks are built from the ground up to match your exact requirements, integrate seamlessly with your existing systems, and provide full ownership without vendor lock-in.

What We Do:

We design and build custom software frameworks for specific use cases including test automation, log management, network monitoring, CI/CD pipelines, data processing, and business process automation. You own the complete source code and intellectual property.

Who It's For:

Organisations requiring specialised automation, unique operational tools, integration platforms, or custom frameworks that commercial products cannot adequately address. Particularly valuable for telecom operators, large IT operations, research institutions, and organisations with unique regulatory or technical requirements.

2. KEY DELIVERABLES

- ✓ Custom-built framework or automation platform
- ✓ Complete source code and intellectual property rights
- ✓ Technical documentation and architecture diagrams
- ✓ Deployment scripts and CI/CD pipelines
- ✓ Training materials and knowledge transfer
- ✓ Optional: Ongoing managed service and enhancements

Test Automation Frameworks

2.1 Key Features

Custom Test Engines

Domain-specific test automation (telecom, healthcare, finance)

Reporting

Custom dashboards, trend analysis, executive summaries

Protocol Support

SIP, Diameter, HTTP, WebSocket, custom protocols

CI/CD Integration

Jenkins, GitLab, GitHub Actions, Azure DevOps

Data-Driven Testing

Test data management and parameterisation

Parallel Execution

Distributed test execution for faster results

Log Management Frameworks

2.2 Key Features

Custom Collectors

Syslog, file tails, API polling, database queries

Analysis

Anomaly detection, pattern matching, alerting rules

Parsing Engines

Regex, Grok patterns, custom parsers for any log format

Visualisation

Custom dashboards using Grafana, Kibana, or bespoke UI

Storage

Time-series databases, Elasticsearch, PostgreSQL, cloud storage

Retention

Configurable retention policies and archiving

Network Monitoring & Observability

2.3 Key Features

Custom Monitoring Agents

SNMP, NetFlow, API polling, packet capture

Topology Mapping

Automatic discovery and relationship mapping

Metric Collection

Infrastructure, application, and business metrics

Performance Analytics

Capacity planning, trend analysis, forecasting

Alerting Engine

Rule-based and ML-powered anomaly detection

Multi-Vendor Support

Vendor-neutral framework supporting diverse equipment

CI/CD Observability & Data Processing

2.4 & 2.5 Key Features

CI/CD Observability & DevOps Automation

Pipeline Frameworks Custom build, test, and deployment pipelines	Infrastructure as Code Terraform, CloudFormation, Ansible automation	Secret Management Secure credential storage and rotation
Artifact Management Binary repositories, versioning, promotion	Environment Management Dynamic environment provisioning	Compliance Checks Automated security scanning, policy enforcement

Data Processing Pipelines

ETL Frameworks Extract, transform, load custom data pipelines	Stream Processing Real-time data processing (Kafka, kinesis)	Batch Processing Scheduled batch jobs with dependency management
Data Validation Quality checks, schema validation, error handling	Orchestration Workflow engines for complex data pipelines	Monitoring Pipeline health monitoring and alerting

Business Process Automation

2.6 Key Features

Workflow Engines

Custom business process automation

Integration Hub

Connect disparate systems via APIs, file transfers, databases

Rules Engine

Business logic execution without code changes

Scheduling

Cron-like scheduling with dependencies and error handling

Audit Trail

Complete logging of all automated actions

Notifications

Email, Slack, Teams, SMS, webhooks

3. KEY BENEFITS

Avoid Vendor Lock-In

Own the complete source code and IP, giving you freedom to maintain, enhance, or migrate without dependency on any vendor.

Perfect Fit for Your Needs

Custom-built to your exact requirements rather than compromising with generic commercial products.

Reduce Long-Term Costs

Eliminate recurring license fees. One-time development cost typically pays for itself within 2-3 years.

Seamless Integration

Built to integrate with your existing systems using your preferred technologies and protocols.

Scalability and Performance

Optimised for your specific workload and scale, not bloated with features you don't need.

Competitive Advantage

Unique capabilities that competitors using standard tools cannot replicate.

Full Control

Modify, enhance, or extend the framework anytime without waiting for vendor roadmaps.

Compliance and Security

Built to your exact security and compliance requirements from the ground up.

4. HOW THE SERVICE WORKS

Phases 1-4: Discovery to Development

Phase 1: Discovery & Requirements (3-4 weeks)

- Understand current manual processes or system limitations
- Document functional and non-functional requirements
- Analyse integration points with existing systems
- Define success criteria and acceptance tests
- Review similar frameworks (commercial and open-source)
- Estimate effort and create project plan

Deliverable: Requirements Specification (50-100 pages)

Phase 2: Architecture Design (2-3 weeks)

- Design system architecture and component diagrams
- Define technology stack and dependencies
- Create data models and API specifications
- Plan deployment and scaling strategy
- Identify risks and mitigation approaches
- Prototype critical components (if needed)

Deliverable: Architecture Design Document (30-50 pages)

Phase 3: Development - Phase 1 (6-10 weeks)

- Build core framework components
- Implement essential features (80% of value)
- Develop unit and integration tests
- Create initial documentation
- Weekly demos to stakeholders

Deliverable: Working Alpha Version (core functionality)

Phase 4: Development - Phase 2 (4-6 weeks)

- Complete remaining features
- Performance optimisation and tuning
- Security hardening and penetration testing
- Comprehensive test coverage (>80%)
- User interface refinement

Deliverable: Beta Version (feature-complete)

4. HOW THE SERVICE WORKS

Phases 5-7 & Ongoing Support

Phase 5: Testing & Validation (2-3 weeks)

- User acceptance testing (UAT)
- Load and stress testing
- Security assessment
- Documentation review and finalisation
- Bug fixing and refinement

Deliverable: Production-Ready Framework

Phase 6: Deployment & Handover (2-3 weeks)

- Production deployment and configuration
- Knowledge transfer (training sessions)
- Documentation handover (technical and user guides)
- Runbook creation for operations team
- Source code handover via Git repository

Deliverable: Deployed Framework + Complete Documentation

Phase 7: Post-Deployment Support (Optional, 30-90 days)

- Hypercare period with enhanced monitoring
- Bug fixes (covered under warranty)
- Performance tuning based on production usage
- User support and troubleshooting

Deliverable: Stable Production System

Ongoing: Managed Service & Enhancements (Optional)

- Ongoing framework hosting and management
- Feature enhancements and improvements
- Dependency updates and security patches
- 24/7 support for critical issues

Deliverable: Maintained and Evolving Framework

5. SERVICE LEVELS & SUPPORT

Project Delivery

- Sprint Cycle: 2-week sprints with end-of-sprint demos
- Communication: Daily standups (team), weekly stakeholder updates
- Documentation: Continuous (not saved for end of project)
- Deployment: Staged (dev → test → UAT → production)

Warranty Period (90 Days Post-Deployment)

- Bug Fixes: Defects in developed code fixed at no charge

Response Times:

- Critical bugs: 4-hour response, 24-hour fix target
- High priority: Next business day response, 3-day fix target
- Medium/Low: 3-day response, agreed timeline
- Exclusions: Issues caused by changes made by customer, infrastructure problems, third-party service outages

Quality Commitments

- Code Quality: 80%+ test coverage, static analysis passing
- Performance: Meets agreed performance SLAs from specification
- Security: No critical vulnerabilities at handover
- Documentation: Architecture, API docs, user guides, runbooks
- Training: Minimum 8 hours knowledge transfer included

6. TECHNICAL REQUIREMENTS

Customer Requirements

During Discovery:

- Clear description of problem or use case
- Access to current systems for integration analysis
- Subject matter expert availability (8-16 hours over 4 weeks)
- Sample data for testing (anonymised if sensitive)

During Development:

- Access to development/test environments
- Stakeholder availability for sprint demos (2 hours every 2 weeks)
- UAT resources (2-4 people for 2-3 weeks)
- Timely feedback on deliverables (48-72 hours)

For Deployment:

- Production environment specifications
- Deployment access (SSH, cloud console, etc.)
- Monitoring and logging infrastructure
- Backup and disaster recovery procedures

Post-Deployment:

- Operations team for knowledge transfer (8-16 hours)
- Ongoing framework administrator designation
- Change management process alignment

TestFyra Technical Capabilities

Programming Languages

- Primary: Python, Go, Java, JavaScript/TypeScript
- Secondary: Bash, PowerShell, Rust, C#
- Specialized: Lua (network devices), Groovy (Jenkins)

Frameworks & Tools

- Web: FastAPI, Django, Express.js, Spring Boot
- Data: Apache Kafka, Apache Spark, Airflow, Pandas
- Testing: Pytest, JUnit, Jest, Selenium, Cypress
- DevOps: Docker, Kubernetes, Terraform, Ansible, Jenkins
- Monitoring: Prometheus, Grafana, ELK Stack, custom solutions

Databases

- Relational: PostgreSQL, MySQL, SQL Server
- NoSQL: MongoDB, Redis, Cassandra
- Time-Series: InfluxDB, TimescaleDB, Prometheus
- Search: Elasticsearch, OpenSearch

Cloud Platforms

- AWS, Azure, Google Cloud Platform
- Serverless architectures (Lambda, Functions, Cloud Run)
- Managed Kubernetes (EKS, AKS, GKE)

Integration Methods

- APIs: REST, GraphQL, gRPC, WebSocket
- Messaging: RabbitMQ, Kafka, AWS SQS, Azure Service Bus
- Files: SFTP, S3, Azure Blob, NFS
- Databases: Direct SQL, ETL tools, CDC (change data capture)
- Protocols: SNMP, Syslog, HTTP/HTTPS, MQTT, custom TCP/UDP

7. SECURITY, COMPLIANCE & CONSTRAINTS

Security Measures

- Secure Development: OWASP Top 10 mitigation, security code reviews
 - Dependency Scanning: Automated vulnerability scanning (Snyk, Dependabot)
 - Secrets Management: Never hard-coded, use HashiCorp Vault or cloud-native
 - Authentication: Support for LDAP, SAML, OAuth 2.0, API keys
 - Authorization: Role-based access control (RBAC)
 - Audit Logging: Comprehensive logging of all actions
 - Encryption: TLS transit, AES-256
-

Compliance

- GDPR: Privacy by design, data minimization
 - ISO 27001: Secure development practices aligned with standard
 - Accessibility: WCAG 2.1 AA for web interfaces (if applicable)
 - Open Source Licenses: Clear documentation of all dependencies and licenses
-

Intellectual Property

- Customer Ownership: You own 100% of developed source code and IP
- License: Apache 2.0, MIT, or proprietary license (your choice)
- Source Code: Delivered via Git repository (GitHub, GitLab, Bitbucket, or self-hosted)
- Documentation: All documentation provided in editable formats
- TestFyra Reuse: We retain right to use generic patterns and techniques (but not your specific code) in other projects

7. SECURITY, COMPLIANCE & CONSTRAINTS

Data & Backup

- Source Control: Git with daily backups
 - Code Hosting: GitHub, GitLab, or your preferred platform
 - Backup Retention: Project files retained for 12 months post-delivery
 - Data Residency: Development performed in UK
-

7. SECURITY, COMPLIANCE & CONSTRAINTS

Service Constraints

Dependencies

- Clear, stable requirements reduce development time and cost
- Access to existing systems required for integration testing
- Timely feedback essential for iterative development
- Quality depends on quality of requirements and test data provided

Limitations

- Cannot develop on proprietary platforms without licenses
- Cannot guarantee compatibility with future versions of third-party systems
- Performance depends on underlying infrastructure capacity
- Cannot reverse-engineer closed-source systems for integration

Not Included (Unless Specified)

- Ongoing hosting infrastructure (unless managed service purchased)
- Third-party software licenses
- Hardware or cloud infrastructure costs
- Data migration from old systems
- Change management and user training
- Regulatory compliance consulting

7. SECURITY, COMPLIANCE & CONSTRAINTS

Service Constraints

Lead Times

- Project kick-off: 2-3 weeks from contract signature
- Discovery phase: Can start within 1 week
- Development start: After discovery phase completion
- Urgent projects: 1-week notice possible (subject to 20% premium)

7. SECURITY, COMPLIANCE & CONSTRAINTS

Onboarding & Offboarding

Onboarding (2-3 weeks)

- Project kick-off meeting
- Discovery workshops (if not already completed)
- Access provisioning (systems, environments, tools)
- Team introductions
- Project governance setup (meetings, communication)

Offboarding (2-3 weeks)

- Final code delivery via Git repository
- Knowledge transfer sessions (minimum 8 hours, can be more)
- Technical documentation (architecture, API, deployment)
- Operations documentation (runbooks, troubleshooting)
- Admin/user training (if applicable)
- 90-day warranty for bug fixes
- Optional transition to managed service

CONTACT INFORMATION - TESTFYRA LTD

Registered Office:

11 Dormer Place
Royal Leamington Spa
CV32 5AA
United Kingdom

General Enquiries:

Email: contact@testfyra.com
Phone: +44 7814 154655
Website: www.testfyra.com

Sales & Commercial:

Email: customer@testfyra.com,
gurjit.badesha@testfyra.com

Company Details:

Company Registration number – 08586825
VAT number - 165527005

This Service Definition Document should be read in conjunction with TestFyra's Terms & Conditions, SFIA Rate Card, and Pricing Document for complete service details.

Version Control: Version 1.0 - December 2025 - Initial G-Cloud 15 submission

END OF SERVICE DEFINITION