

Terms of Reference
Development of Mobile Applications
ICTA/GOSL/CON/QCBS/2016/134

1. Introduction;

“Mobile Application Development” is a linked initiative with the project - 'Lanka Gate', the Service Oriented Architecture (SOA) based messaging infrastructure, which is envisioned to be the gateway for electronic information and electronic delivery in Sri Lanka. The 'Lanka Gate' project is envisioned by the e-Sri Lanka initiative, and also stated in the e-Government policy document approved by the Cabinet of Ministers, that practically all the electronic services (eServices) and electronic information in Sri Lanka will be delivered via Lanka Gate. (Refer: <http://www.icta.lk/en/programmes/re-engineering-government/131-main-projects/556-the-lanka-gate-initiative.html>)

With a phenomenal growth of mobile phone and internet subscribers in Sri Lanka (Mobile Subscribers – 2016 Mar: 24,568,745 /Mobile Subscription per 100 people – 2015 Dec: 116.7 / Mobile Broadband Subscribers – 2016 Mar: 3,484,079), mobile technology is revolutionizing the way the service providers interact with end-users, whether the end-user is a customer, employee, business partner, or even a service or smart device.

Hence the mobile application development project is steadily targeting to address the need of converting key identified services into the applications on mobile followed by a priority stack.

2. Background;

In order to rapidly innovate and remain competitive in a mobile society, an organization must have a comprehensive mobile app that reaches its individual target segments. With the increasing use of smart phones and mobile devices, mobile applications have been recognized as a key service baseline.

By putting the understanding in to practice, ICTA commenced developing mobile applications to deliver government services to citizens in the most effective and proficient manner. The mobile application for train schedules was the kick-off under this segment which was developed by ICTA based on the existing eService offered through country portal. At the moment the Government mobile app base consist of several services offered by different government organizations.

Since it is one of the most convenient mechanisms to access a range of services through mobile devices, ICTA intends to initiate new projects aiming the implementation of a range of new mobile applications, which will cater wide scope of requirements.

3. Concise statement of the objectives;

ICTA intends to procure and obtain the services of a consultant firm to implement selected set of new mobile applications for Government Organizations. The consultant firm is required to gather requirement, design and develop mobile applications and ensure smooth deployment targeting to deliver government informational service to citizens through mobile devices.

4. Scope of Work;

(a) Implementation approach;

- 4.1 ICTA has developed number of eServices offered by various government organizations which are currently available through the country portal. ICTA intends to develop mobile applications for selected eServices enabling citizens to obtain the services through mobile devices.
- 4.2 In addition to the already developed eServices, ICTA has identified potential information services offered by various government organizations which can be delivered through mobile devices. ICTA also intends to develop mobile applications for selected services.
- 4.3 The consultant firm may use existing web services for mobile application development for already developed eServices
- 4.4 For other potential services, the consultant firm may develop the mobile application with the backend web application to retrieve and manage content with the government department
- 4.5 Currently, the Government mobile application range (Android based) – hosted at Google Play under the name “ICTA” consist of native mobile apps (Android) with services offered by government organizations. The consultancy firm may require to convert these apps into hybrid mobile apps
- 4.6 Therefore once this contract is awarded, the consultancy firm, in collaboration with ICTA, is able to select the implementing mobile applications from the available list of existing eServices, existing mobile apps (for conversion into hybrid apps), and the potential new services.
- 4.7 Each selected mobile app development will be considered as a sub-project.
- 4.8 The consultant firm is required to undertake at least 2 mobile application developments at the same time (sub-projects). Therefore should ensure adequate development team strength.
- 4.9 Key consultants are required to be available on-site as and when needed by the respective sub-project.
- 4.10 At the inception phase of each mobile application development, the consultant firm is required to submit the number of Person days along with DSRS and other documents. The value of a given mobile application development is calculated based on the Person day rate.
- 4.11 All consultants are required to sign a Non-Disclosure Agreement (NDA) where applicable.
- 4.12 The consultant firm is required to manage each mobile application service for one year period from the date of launch.

(b) Scope of Work;

Therefore the scope of work can be listed as indicated below;

- 4.13 Identify mobile applications to be implemented in collaboration with ICTA.
- 4.14 Conduct a system study of the potential mobile application.

- 4.15 On completing the above, Submit a Proposal comprising of the following, among others;
 - 4.15.1 Requirement specification of the mobile application
 - 4.15.2 Number of person days of the assignment
 - 4.15.3 Mobile application development schedule
 - 4.15.4 User Acceptance Test (UAT) Criterion
- 4.16 The above Proposal should include all deliverables as specified in below item '5 – Deliverables and time lines'
- 4.17 Implement mobile application, upon obtaining ICTA approval for the above.
- 4.18 All the mobile applications developed should support Android, iOS and Windows phones platforms (Hybrid mobile application development) and under each platform, should be cross version compatible.
- 4.19 The consultant should purchase any licenses/publishing materials required to develop or deploy mobile applications on behalf of ICTA. At the end of the assignment all licenses/publishing materials with the source code should be transferred to ICTA
- 4.20 The consultant should develop a backend application for mobile apps which developed for potential government organizations
- 4.21 Adherence to open standards and Service Oriented Architecture (SOA) principles. (If back end application is developed)
- 4.22 Adherence to e-Government Policy of Sri Lanka [2].
- 4.23 Adherence to LIFe standards [3].
- 4.24 Maintain project source code in the ICTA Source Code Management system (SCM).
- 4.25 Maintain all issues in the Issue tracking system maintained by ICTA.
- 4.26 Adopt a proper application release procedure to release the mobile applications
- 4.27 Participate for Project Review Committee meeting and Project Implementation Committee (PIC) Meetings as a member
- 4.28 Obtain User Acceptance (UAT) for the implemented mobile application.
- 4.29 Implement the mobile applications in collaboration with the SQA consultants appointed by ICTA, or review committee and facilitate the 'Software Project Audit Process' specified by ICTA. Refer Annex 4.
- 4.30 Integrate with Internet Payment Gateway Systems as decided(for the mobile apps that are having a payment component)
- 4.31 Provide sufficient training to government organizations on the mobile application and back end handling as required
- 4.32 Work collaboratively with ICTA and government organizations throughout the tenure of the project duration.
- 4.33 Refer following Annexes which form a part and parcel of the Terms of Reference.

Annex 1 - Potential Mobile Applications

Annex 2 - Non-Functional Requirements

Annex 3 - The Lanka Gate Initiative - Overall Architecture & Design

Annex 4 - Software Project Audit Process

5. Final outputs, Reporting Requirements, Time Schedule for Deliverables;

Project duration shall cover a maximum ceiling of 300 Person Days including requirement gathering, designing, developing and deployment of mobile applications.

It is expected that the consultancy firm will implement as many sub-projects as possible, provided that the accumulated total number of man days is within the total estimated number of person days for this assignment (300 Person Days).

Consultancy firm is required to submit the following list of deliverables for each sub-project (Government Organization wise mobile application implementation project).

No	Deliverables	Phase
5.1	Mobile Application Implementation Proposal 5.1.1 Requirement specification of the mobile application 5.1.2 Number of person days of the assignment 5.1.3 Mobile application implementing schedule 5.1.4 QA Plan and Test Cases 5.1.5 Operational and back office support requirement (if the backend application to be developed) 5.1.6 Acceptance criteria for Deliverables and UAT 5.1.7 Proper maintenance of issues in the ICTA Issue tracking System 5.1.8 Specifications for devices if required (Eg. Mobile devices)	Inception
5.2	5.2.1 Design document 5.2.2 Release Management plan (including staging and production) 5.2.3 Data migration and integration plan (if applicable) 5.2.4 Proper maintenance of issues in the ICTA Issue tracking System	Elaboration
5.3	5.3.1 Proper maintenance of source code in SCM 5.3.2 Proper maintenance of issues in the ICTA Issue tracking System	Construction
5.4	5.4.1 Solutions installation guide 5.4.2 User manual (if the backend application to be developed) 5.4.3 Administrator manual (if the backend application to be developed) 5.4.4 Proper maintenance of issues in the ICTA Issue tracking System 5.4.5 Government organization level training (on the back end application – if any, mobile application – if required) 5.4.6 Successful UAT acceptance of the Mobile Application and deployment	Transition

Refer http://en.wikipedia.org/wiki/IBM_Rational_Unified_Process for more information about RUP (Rational Unified Process) phases.

6. Services and Facilities Provided by ICTA

- 6.1 Web-based access to the ICTA SCM system
- 6.2 Access to staging/ production servers
- 6.3 Issue Tracking System
- 6.4 Available source code of the relevant eservice/s

7. Review Committees and Review Procedures

All versions of deliverables will be reviewed by the team appointed by ICTA

References:

[1] Constructive Cost Model - <http://en.wikipedia.org/wiki/COCOMO>

[2] e Government Policy Approved By Cabinet of Sri Lanka - <https://www.icta.lk/icta-assets/uploads/2016/03/eGov-Policy-structured-v4-0.pdf>

[3] Lanka Interoperability Framework - <http://www.life.gov.lk/>

– END –

Potential Mobile Applications

Potential Mobile Applications
<p>1.) Department of Motor Traffic <i>Service 1: Revenue License Issuance Service</i> <i>Service 2: Revenue License active/expiry status</i></p> <p>2.) Department of Examination <i>Service 1: View Exam Results</i></p> <p>3.) Department of Wildlife Conservation <i>Service 1: Reservation of Bungalows at Wildlife parks</i> <i>Service 2: Bungalows Reservation Status Enquiry</i></p> <p>4.) Colombo Municipal Council <i>Service 1: Property Tax (Rates) payments and balance enquiry</i> <i>Service 2: Trade Tax (Taxes imposed for trades) payments and balance enquiry</i> <i>Service 3: Payments for Tax on businesses (e.g. special businesses like Pawn shops) and balance enq.</i> <i>Service 4: Market Rental (Rentals for market places) payments and balance enquiry</i> <i>Service 5: House Rental payments and balance enquiry</i> <i>Service 6: Shops and Boutiques Rental payments and balance enquiry</i> <i>Service 7: Hawkers Rental (rentals for very small shops) payments and balance enquiry</i> <i>Service 8: Balance Enquiry of Different Tax Types</i></p> <p>5.) Department of Police <i>Service 1: Issuance of Police Clearance Certificate</i> <i>Service 2: Clearance Certificate Application Status Enquiry</i></p> <p>6.) Department of Forrest <i>Service 1: Online Reservation System for Circuit Bungalow Booking</i> <i>Service 2: Online Ticketing System for Wild Life Safari</i></p>
High-Level System View (Potential)
<pre> graph LR subgraph Backend A[Department Backend Application] -- A --> B[ESB] B -- A --> C[eService Hosted in Country Portal as a Web App] end C -- A --> D[Mobile Application] </pre> <p>Some of the e-services are currently available through county portal (www.gov.lk)</p> <ul style="list-style-type: none"> Restful Web services (Above A) are used to retrieve data from the backend application/s at different levels Same web services will be securely exposed to the proposed mobile application through the ESB Any new web service that is required to implement the mobile application will have to be implemented along with the mobile application and shall be securely exposed to the proposed mobile application through the ESB

Non-Functional Requirements

While developing and testing mobile applications for Lanka Gate related eServices, you are required to maintain the following common set of Non Functional Requirements (NFR). Though any specific boundaries are given, you are required maintain the given requirements in industry acceptable levels.

Non Functional Requirement	Description
Install and Launch	If the application takes a long time, need to ensure to notify the user
	Time to un-install – The application must uninstall from the device
	The application must not download any additional code
Memory Usage	Ensure that the application correctly handles out of memory exceptions during application execution
	Suspend and re-launch of the application is handled properly
	The application recovers from an idle state correctly
Connectivity	When the application uses network capabilities, it must be able to handle network delays and any loss of connection
	When the Application uses network capabilities, it must be able to handle the device being in Airplane mode
Event Handling	Ensure that the Application behaves correctly on expiry of a timed event while the Application is running
	Ensure that the Application resumes correctly from a suspended state on expiry of a timed event
	Ensure that the Application starts correctly from an exited state on expiry of a timed event.
Interruptions	Check that the application continues to function through interruptions
Display	Correct screen repainting <ul style="list-style-type: none"> • The Application screens must be correctly repainted, including cases when edit boxes and dialog boxes are dismissed • There must be no blinking of moving objects and background. If the Application objects overlap they must still render correctly
	UI Consistency <ul style="list-style-type: none"> • The Application UI should be consistent and understandable throughout, e.g. common series of actions, action sequences, terms, layouts, soft button definitions and sounds that are clear and understandable
	Key Layout – Ease of Use <ul style="list-style-type: none"> • The buttons should be easy to use. • Button usage should be suitable for both a left-handed and right-handed person, within the physical constraints of the device design.
	Application Speed <ul style="list-style-type: none"> • The Application works in the device it was targeted for, and it is usable on the device: the speed of the Application is acceptable to the purpose of the Application and must not alter the user experience by being uncontrollable.
	Error Messages <ul style="list-style-type: none"> • Any error messages in the Application must be clearly understandable • Error messages must clearly explain to a user the nature of the problem, and indicate what action needs to be taken (where appropriate), it should not display technical terms and should be understandable by general public
	Actions while Rendering <ul style="list-style-type: none"> • Application must not perform inappropriate actions while thinking or rendering
	Different Screen Sizes <ul style="list-style-type: none"> • Where the application is designed to work on multiple devices it must be

	<p>able to display correctly on differing screen sizes</p> <p>Multiple Display Format Handling -</p> <ul style="list-style-type: none"> Where the device and Application can display in multiple formats (e.g. portrait / landscape, internal / external display), the elements of the application should be correctly formatted in all display environments. <p>Font adjustment – The text font size may need to be adjusted up or down to keep the text readability</p> <p>Layout tweaks – The layout may need to be adjusted to increase or decrease the spacing between and around labels and widgets shown on the screen</p> <p>Image changes – Background images or background art may have to be provided in two different versions. A large size/high resolution version and a small size/low resolution version so that it properly fills the amount of physical space available on the screen</p>
Performance	<p>iOS Performance Measurement Suite https://developer.apple.com/library/mac/#documentation/performance/Conceptual/PerformanceOverview/PerformanceTools/PerformanceTools.html</p> <p>Android Performance Best Practices https://developer.android.com/training/best-performance.html</p> <p>Microsoft Windows Phone Application Analysis http://msdn.microsoft.com/en-in/library/windowsphone/develop/jj215908(v=vs.105).aspx</p> <p>iOS The XCode development environment comes with a rich suite of performance measurement, analysis and monitoring tools such as Instruments, Shark, Activity Monitor and BigTop which allows to measure and monitor performance.</p> <p>Android Android Developer Tools (ADT) comes with the Profiler tool which can track memory performance. Also HPROF tool to look at memory usage on Android (hprof output can be generated and analysed in a variety of different ways. See Android developer documentation for details). Several third party applications are also available on the Play Store to help measure and analyse CPU, memory and battery performance of the application.</p> <p>Windows Phone The Windows Phone Application Analysis tool available in Visual Studio to perform analysis CPU, memory and battery performance.</p>
Keys	<p>Scrolling in menus - The menu item list with no adverse effects on the Application</p> <p>Text field scrolling - This should scroll vertically and (if applicable) horizontally in the dialog.</p> <p>Simultaneous key presses - Ensure that the Application copes with simultaneous key presses or multiple touch</p>
Stability	<p>Application behavior after a forced close - The Application must not crash or freeze at any time while running on the device.</p>
Privacy and User Permissions	<p>Privacy - The app must state and comply with a Privacy Policy and Privacy best practice</p> <p>If the application holds or accesses personal data there must be an associated privacy statement.</p>

	<p>The application must obtain permission from the user to send, share or store any personal data.</p> <p>The application should work without requiring private data to be shared.</p> <p>The use of personal data must be part of the principle function of the application, and the user's consent for any additional uses must be explicitly obtained.</p>
	<p>Location Permissions - The app must ask for permission to use location data The application must notify the user before collecting, storing or transmitting the device location data.</p> <p>Use of the location data must be relevant to the function of the application.</p> <p>The use of location for targeted advertising must be explicitly stated or the application must obtain the users permission.</p>
Network	<p>Support for different network channels - (WiFi, GPRS, 2G/3G/4G)</p> <p>The application must not use excessive network resources</p> <p>Signal drop or signal strength reduction - Evaluate each network-enabled feature in the application in the situation where the channel over which it is operating becomes unavailable or its signal strength reduces.</p>
Battery Usage	<p>Below three main battery power tests must be performed to ensure the the application battery usage.</p> <p>Normal use test Start on a full battery and use the application for 6-12 hours and measure the battery level at the end of each ½ or 1 hour. An automated testing tool can be used to keep the test running for the required time interval. This test results will explains how quickly the application drains the battery when in 'normal' use, with all the foreground and background features of the application running normally.</p> <p>Idle run test Turn off the screen lock and power saver modes on the device. Then start on a full battery and keep the application running on its main, home or dashboard screen as appropriate, and measure the battery level at ½ or 1 hour intervals. This test will measure the battery drain due to such things as intentional or unintentional automatic screen refreshes, and due to the background threads or services running in your application.</p> <p>Screen lock test Perform test (2) again but with the device screen in locked mode. This will allow testing if the application is consuming any CPU and/or network resources (and hence the battery) when it's not viewable to the user. In this case, to avoid unnecessary battery drain, any kind of screen activity such as automatic screen refreshes and also background thread or services in this mode should be paused / stopped.</p>
Operating System	<p>Should support three major previous versions of the particular OS</p> <ul style="list-style-type: none"> • Avoid extensive use of third party libraries • Avoid use of deprecated APIs • Use best practices in using APIs and avoid non-standard usage of API methods • If using features specific to an OS, ensure that they degrade or fail gracefully on previous OS versions.
Security	<p>Confidentiality and Integrity All developed mobile applications/ back end e-services should ensure</p>

	<p>“confidentiality” and “integrity” whenever required by adhering to transport and message level security standards. (i.e. HTTPS, WS-Security)</p> <p>Encryption - All sensitive information (personal data, credit card & banking information etc.) must be encrypted during transmission over any network or communication link.</p> <p>Passwords - Passwords and sensitive data are not stored in the device and not echoed when entered into the App, sensitive data is always protected by password.</p> <p>Availability All mobile applications / back end e-services should be developed to ensure “High Availability” to remain the system available all the time. (e.g. mobile applications clustering capability should be taken into consideration in the development)</p> <p>Non-repudiation All mobile applications / back end e-services should ensure non-repudiation by having standard audit-trails and provisions to have WS-Security using digital signatures.</p>
--	--

[ANNEX 3]

THE LANKA GATE INITIATIVE - OVERALL ARCHITECTURE & DESIGN

(a) Introduction to Lanka Gate

As an important component of the e-Sri Lanka initiative, it is envisioned that practically all the eServices and electronic information in Sri Lanka will be delivered via a comprehensive integration platform. This wide collection software infrastructure and systems which is envisioned to be the gateway for electronic information and electronic interactions in Sri Lanka, is generally referred to as the 'Lanka Gate' initiative.

Many eServices will be generated as a result of various projects done at the ICT Agency, such as the Population Registry project, the ePensions project and the Samurdhi Services project. In addition, many other eServices could be generated by government, public and private sector organizations as well as by community groups. Lanka Gate would include a comprehensive collection of infrastructural mechanisms to easily 'plug-in' an eService or to 'compose' a set of eServices in order to generate an composite eService, such that these eServices would be readily and easily available to other applications and portals that comprise Lanka Gate. For this purpose, it is envisioned that the projects within Lanka Gate would be designed to leverage Web 2.0 concepts, open standards and a Service Oriented Architecture (SOA), enabling dynamic, customizable, collaborative and compose-able services via multiple delivery channels.

Thus the collection of software systems that comprise Lanka Gate would collectively provide an *enabling infrastructure for rapid integration and delivery of eServices*, leveraging loosely-coupled

architectural principles to encourage the creation of innovative applications, solutions, and business models, communication models, pricing models and service mash-ups by various stakeholders across the country.

The intention is that this architectural blueprint will guide the various software engineering projects that would eventually be integrated into Lanka Gate. Since Lanka Gate will always be in a state of flux with the continuous addition of eServices from new projects, removal of old eServices as well as the generation of new applications, portals or composite eServices via services mash-ups or services composition, it is hoped that this overall architectural blueprint would continue to 'live' as a vision of what the end result should embody. Furthermore, it is expected that the launch of the Lanka Gate initiative will be coupled with the roll-out of a strong SOA Governance Model.

(b) Lanka Gate: The Core Components

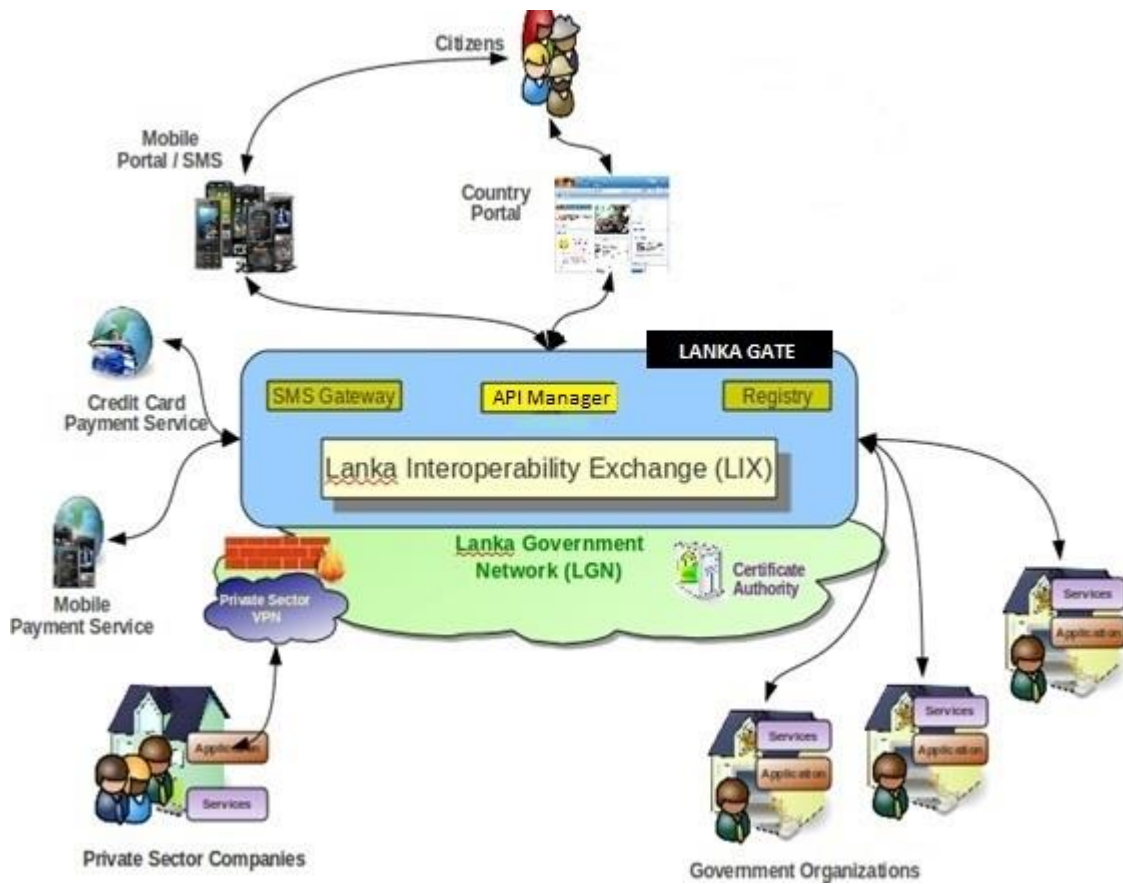


Figure 1 – The Conceptual Architecture

The conceptual design shown above in Figure 1 illustrates the loosely-coupled and flexibility of the Lanka Gate infrastructure. It is composed of following core components.

1. Lanka Interoperability Exchange Project (LIX)

The Lanka Interoperability Exchange (LIX) delivers all the interconnectivity and discovery capabilities that services implemented by the various projects need, by facilitating message routing, transport management, transaction management, mediation, transformation, policy enforcement and service discovery. As an example, considering the eGovernment domain, the LIX would provide the fundamental capabilities necessary for government-wide services to efficiently achieve the vision of re-engineering government in Sri Lanka. Likewise, considering the eCommerce domain, the LIX would enable businesses to create revenue-generating models that would be able to innovatively utilize the infrastructural interconnection capabilities of the LIX to consume the eServices.

LIX is built on top of an Enterprise Service Bus (ESB). It therefore harnesses ESB capabilities such as routing, mediation, messaging, service orchestration and management of eServices and allows the use of a wide range of open protocols and open standards such as JMS, SMTP, XMPP, CORBA, REST and SOAP to connect existing and new systems/services.

In addition to providing message transport related services, the LIX also provides service discovery capabilities and features a collection of important authentication and authorization related eServices that would facilitate business & e-government transactions which require higher levels of security.

Thus the LIX and its associated protocols create an enabling framework that provides a secure, trusted channel through which government, public and private sector organizations may communicate and transfer information amongst each other. The LIX enables organizations to offload common functions such as authentication, authorization and payment, thereby allowing them to focus on business or domain specific functions. By providing such a shared infrastructure reduces the cost of implementation, enabling organizations to rapidly innovate and implement eServices that they otherwise may not even have considered. End users benefit from this shared infrastructure as it drives consistency in the way services are delivered compressing the user adoption and learning curves.

Conceptually, the capabilities offered by LIX are aligned with the enterprise computing notion of *integration-as-a-service (IAAS)* where businesses access a single hub that interconnects all trading partners, facilitated by SOA.

2. Country Portal (CP)

The Country Portal (www.gov.lk) serves as a primary web interface that connects users to the eServices provided within the Lanka Gate concept. Thus the Country Portal is a fundamental access point for citizens, non-citizens, businesses, agents and government employees to various government organizations and businesses in Sri Lanka. The Country Portal features multiple service delivery channels to accommodate various end user realities.

The Country Portal project is a container which provide access to eServices Web application which are self-contained front-end interfaces to either a single eService, several eServices from a specific project, or a transactional/mashup combination of eServices across several projects.

The web browser based delivery channel of the Country Portal features a highly user-friendly, dynamic interface, providing the end-user with the capability to design their own interactive user experience based on their particular needs and preferences. Most of the Web 2.0 capabilities available in Lanka Gate will be delivered through the web browser based delivery channel.

3. Credit Card On-line payment Services

A system to enable credit card payments for government enabled eServices, thereby facilitating electronic commerce for credit card holders.

4. Mobile Payment Services

A system to enable payment via a mobile phone for government enabled eServices, thereby facilitating electronic commerce for mobile phone users (This is yet to be integrated).

5. SMS Gateway (GovSMS)

A common interface open for mobile service providers to establish in-bound and out-bound Short Messaging Services (SMS) with Lanka Gate architecture. The mobile information and service gateway built as a part of Lanka Gate by ICTA to use the common, short telephone code "1919" should be used by all government organizations for delivery of such information and services.

6. Service Registry

The *service registry* provides the infrastructure to define and manage meta data of the SOA in a well-structured manner. Features such as, access control, version management, tagging, linking, searching, and notification, can be utilized in order to implement the “design-time SOA governance”.

LIX uses this *service registry* as the configuration store as well as the policy store to read policy information associated to each of the service. This is in combination with the monitoring capability of the LIX to formulate the “runtime SOA governance”.

(c) Quick Win e-Services for Lanka Gate

As mentioned above, the Sri Lanka “Country Portal” is an already implemented web application, where citizens are expected to interact with eServices offered by various government departments, for a variety of citizen eServices. The Country Portal closely interacts with the Lanka Interoperability eXchange (LIX) Enterprise Service Bus backbone, to interact with the actual service implementations hosted by the various departments, in a Service Oriented Architecture (SOA) style deployment.

In “**Quick Win**” projects, the government department back offices are not going to be re-engineered fully but will undergo “minimum” business process changes. However, they are going to be important services with a very high impact.

The LIX is already connected with an on-line payment gateway and planning to be integrated to a Mobile Payment gateway. E-services that require payments from citizens are expected to use only these systems for the processing of the payments. An SMS gateway is also connected to LIX, so that citizens could make queries and receive status updates or receive other information from the eServices they utilize.

(d) E-service Development for Lanka Gate

As mentioned above, the eServices to be implemented are NOT expected to implement any major systems or replace any of the existing systems at the various government departments. They are expected to tap into any existing services already implemented, or expose new services as required with minimal disruption and changes to these existing systems. Hence, there can be two basic scenarios that can be envisioned (See Figure 2).

Scenario 1: This is where a minimal changes are required. The considered department consists of a working application with a connected database OR even it may have well-written web services that can be exposed to Lanka Gate. If not, it will be a matter of exposing some according to the requirement.

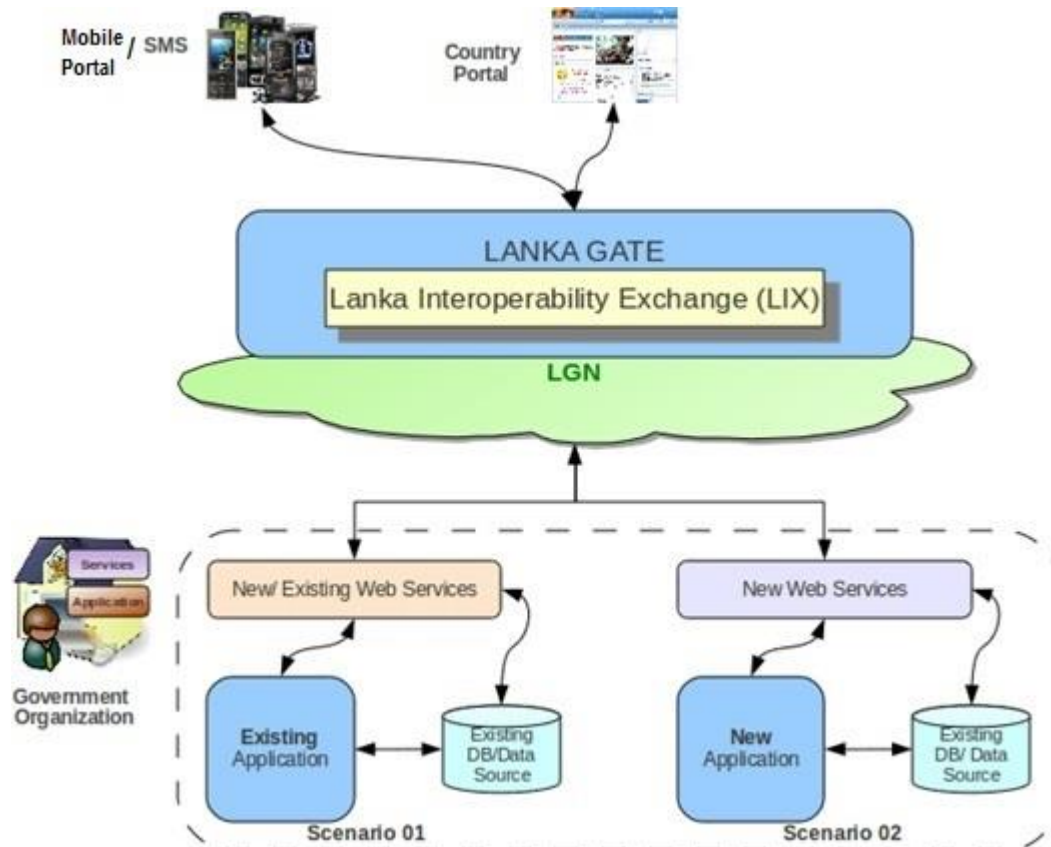


Figure 2: Developing eServices to Lanka Gate

Scenario 2: This is where SOME changes required. If the department only has a data source such as a spreadsheet, it is required to write a new application allowing the data source to be connected to the newly written web services. Otherwise, if the existing DB needs cannot be used directly to a web service, again a new application should be written to bridge the DB with web services. This complexity of this newly created application will depend on the complexity of other back office applications within the department. A proper *Business Process Modeling* (BPM) tool can be leveraged to ease this task depending on this complexity.

However, irrespective of the DB or the data source, it is required to write new web services to expose the back office systems to the Lanka Gate.

Certain eServices may allow the citizens to save information into the new systems, and these systems would require a database for persistence of this information. In addition, certain services may require a citizen to make payments – and these would be facilitated via the mobile or on-line payment gateways, or any existing payment mechanisms used by the department – such as via direct payment to a bank. Thus the back-end support systems would need the ability to interact with the payment gateways and any direct interfaces to bank payment information, to ensure proper payments have been made.

In addition, some of these new systems may require an internal web based system to query information on these new eServices, as well as generate reports etc. To support these use cases, an internal web based application may need to be developed, supporting role based access for use by the internal departmental staff. As an example, if a citizen applies for some facility and electronically submits a set of documents, and makes a payment, the citizen should be able to visit the department with the relevant reference numbers, and a staff officer would then be able to verify the authenticity of the supporting documents, and confirm the payment, so that the facility could then be made available to the citizen with a shorter processing time. In addition, some of these

eServices may allow a citizen to schedule such a visit to the department – to ensure expected levels of service. Hence such a scenario would require the back end system to perform a simple scheduling of the applicants to the department depending on certain variables.

Developing Web Applications for Lanka Gate eServices

For any eService, a simple web application should be developed and these web applications must be able to access via country portal as well as independently via the respective department's web site. The web applications must be able to support English, Sinhalese and Tamil. If the eService is a simple query (e.g. status check), the web application would be able to call into the existing web services or a new web service developed to cater to the use case in question, through a SOAP web service call through the LIX.

Developing SMS Services for Lanka Gate eServices

If the query service in question, is also offered over SMS, the SMS gateway would be able to invoke this same web service, and respond back to the user with the results. Some eServices may allow the user to subscribe to certain events (e.g. change of status, delay of an application etc), at which point, the system should push SMS updated back to the user via the SMS gateway – if the user has specified a mobile number, and requested SMS notifications. When a new SMS is received by the SMS gateway, it will be routed to a SOAP service of the target department, and each department will then have to implement the SMS request processing logic, and optionally response where applicable. Note that unless explicitly specified, all communication through LIX would be SOAP web services calls only.