Living PlanIT

Urban Operating System (UOS™)

Introduction to the Living PlanIT UOS™ Architecture, Open Standards and Protocols.
Table of Contents

Introduction .................................................................................................................................................................... 3
   Urbanization and emerging technologies ........................................................................................................... 3
   The UOS™ in the context of future proof cities .............................................................................................. 4
   Living PlanIT ...................................................................................................................................................... 4
The Urban Operating System (UOS™) .................................................................................................................. 5
   The UOS™ - Overview ..................................................................................................................................... 5
   The UOS™ Architecture .................................................................................................................................. 6
   The Benefits of the UOS™ Architecture ........................................................................................................ 8
   The UOS™ in Practice ..................................................................................................................................... 8
   The UOS™ and the Distributed Cloud Architecture ...................................................................................... 10
Interfaces and Protocols .......................................................................................................................................... 13
   Overview .......................................................................................................................................................... 13
   Architecture .................................................................................................................................................... 13
   UOS™ Application Program Interface (API) ............................................................................................... 14
   UOS™ DSI (Distributed Scale Interface) ..................................................................................................... 15
   UOS™ SCI (Sensor & Controls Interface) .................................................................................................... 16
   Standard Protocols ......................................................................................................................................... 16
   Adaptable Transports ..................................................................................................................................... 17
   Existing Drivers / Legacy Systems ................................................................................................................ 17
   New Drivers .................................................................................................................................................... 20
   UOS™ EAI (External Asset Interface) .......................................................................................................... 21
   Open Specification Philosophy .................................................................................................................... 21
Introduction

Urbanization and emerging technologies

By the year 2050 more than 70% of the world’s population will live in cities consuming the bulk of the world’s resources. The world of government is changing amid increasing pressure to deliver solutions to citizens that are not only economically sound but environmentally and socially responsible. On the other hand, the emergence of Internet services and an ‘always-on’ culture has simultaneously raised expectations – particularly amongst younger citizens. Solutions are urgently required to manage city resources efficiently while also providing a platform for citizen services as a key enabler to economic and social development.

As with other sectors that have benefited from technological innovation, urban development and operations will be significantly improved through the application of converging methods and technologies including:

- Systems thinking – design architecture and interoperability that enables elements of the system to collaborate with others to maximize the use of resources i.e. surplus energy produced from renewable sources stored in water systems and automotive platforms etc.

- Product lifecycle design and management – application of manufacturing methodologies and techniques commonplace in other industries that takes advantage of modeling and simulation to design for future operations which in turn informs and enhances the design and fabrication of complex assets i.e. simulation of urban physics, spatial relationships, materials, human ergonomics, logistics and environment to predict outcomes and make informed engineering and design choices that reduces lifecycle costs and enables the continued augmentation of environments to meet user’s needs.

- The Industrialization of the Internet (sometimes referred to as "Machine to Machine Communication" or "M2M") – The Internet has evolved from simple information sharing, through communication and collaboration and is now entering the era of ubiquitous connected devices. These devices may be simple sensors or actuators, materials, mobile devices such as smart phones or wearable medical appliances through complex machines and infrastructure. Control systems are commoditized and replaced by network hardware that connect, interrogate (sense), analyze and control (actuate) devices, in turn harvesting intelligence that enables continuous improvements in the efficiency and reliability of city and citizen’s services.
The UOS™ in the context of future proof cities

This whitepaper explains the Living PlanIT Urban Operating System (UOS™), the industrial Internet platform and principal infrastructure that is integral to successfully envisioning, retrofitting, building and managing cities in the 21st century (“Smart”, “Intelligent” or “Living Cities”). This includes recognition of the need for continual evolution; the importance of clear metrics and analytics; the increased connection between urban dwellers and the buildings in which they live and work; a sense of possibility and openness; increased efficiency; generative structures that learn; agile infrastructures that serve multiple functions and respond to environmental and other changes; and resilient systems that can recover without breaking down and that resist obsolescence. Many experts and commentators cite Living PlanIT as being the leading contender in providing more considered solutions to the problem of smart urbanization. We thus envision Living Cities as generative, inclusive, agile, dynamically evolving and resilient.

Living PlanIT’s Urban Operating System (UOS™) provides the essential middleware that enables networked sensors and actuators to be deployed at scale, coordinated through a unified and secure real-time control layer which also shares and collects data across the entire urban landscape. This time series data converges with UOS™ spatial analytics to be mined for further insight enabling continual incremental efficiencies and applications that enhance urban services to drive economic, social and environmental development through new forms of human interaction.

The UOS™ also provides a set of data and application services that facilitates the leveraging of building and city facilities and information by applications known as PlaceApps by making them location and context-aware. This enables the application developer community to quickly and simply build applications in that urban context for delivery to citizens, governments, service providers, and real estate developers and operators alike.

This allows cities and its citizens to make better decisions about their use of financial, natural and human resources, thereby creating a more sustainable city and dramatically improving the level of engagement a city has with its citizens and creates the flexibility cities need to make them “future-proof”.

Living PlanIT

Living PlanIT is a technology company that has developed and licenses the Urban Operating System (UOS™) - the essential Machine-to-Machine platform that enables the design, delivery, operations and enhancement of urban and rural environments, facilitating the interaction between people, places and things.
The Urban Operating System (UOS™)

The UOS™ - Overview

The Living PlanIT Urban Operating System (UOS™) is a software platform that accelerates the development and deployment of urban technology and connected devices. It provides software-based flexible real-time control, as well as supervisory control, data acquisition and lifecycle management. Its real-time and historic analytical functions, together with a comprehensive Services Oriented Architecture API, simplify the building and operation of advanced data-driven and control-based applications on a variety of end-user devices.

Living PlanIT primarily focuses the UOS™ on:

- Living Cities / Smart Urban Developments – the emerging market for increasingly intelligent, sustainable buildings and infrastructure, whether deployed in new cities, major regeneration projects in existing cities, or individual developments.

- Machine-to-Machine Communications - the ‘connected device’ market, based on increasingly pervasive networks and providing remote sensing and interaction with a wide array of embedded and mobile devices, ranging from smart-phones to vehicles, to robotic devices, to sensor ‘motes’.

The majority of applications are delivered by a growing ecosystem of partners, within projects in which Living PlanIT is engaged as well as directly through their own channels that look to drive specialized applications targeted at enhancing the design and delivery of smart cities and connected devices.

Living PlanIT supports optimized deployment of the UOS™ through the provision of professional services from Digital Masterplanning, through UOS™ planning and commissioning to Operations and Analytical support. The UOS™ reduces the cost of providing control architectures in an urban context as well as advanced information and interaction management, in both private and public cloud contexts, by making extensive use of standard hardware and software components from partners. Further, by providing a unified platform for all applications, sensors, and data, deeper insights are enabled because more naturally aligned information is available to interpret, in turn...
driving a richer history to provide predictive models and optimization. This means cheaper infrastructure, progressively more efficient operations and lower operating costs, and better consumer and user experiences.

The UOS™ Architecture

The Urban Operating System (UOS™) is sometimes described as the collision of next-generation software-based Building Management Systems, SCADA (Supervisory Control and Data Acquisition) and Cloud computing. To put it another way, it provides a unified sensor data acquisition, real-time control, historical database, analytics engine, and application hosting platform for urban environments, or – deployed in a public cloud – for remote devices with sensing and actuation capabilities.

The UOS™ has been architected to allow flexibility in deployment and also to scale from a small / local controls scope to large scale domains, ultimately comprising entire cities. Examination of the architecture shown below helps to explain how this is achieved.
Broadly the architecture can be seen to have four distinct layers:

**The Sensor/Actuator Network** – a unified, converged network which is enabled by the UOS™ but is not considered part of it. The UOS™ communicates with devices in this layer to collect data, make decisions – sometimes with user input via applications - and issue commands to controllable equipment. In an urban environment the network will typically be a local / metropolitan area network.

**The Controls Layer** – the ‘first layer’ of the UOS™, deployed with network infrastructure. This provides the most distributed point of intelligence in the system, and provides the most time-critical responses to incoming data. It also is where the specific code – or Driver Applications - for interfacing with specific equipment resides if needed. Control Applications also reside at this level – these are managed through the supervisory layer of the UOS™ but run in the UOS™ Real Time Control (RTC™) core to provide autonomous, immediate responses to control requirements, for example the control of a light or a motorized flap that forms part of an HVAC system.

**The Supervisory Layer** – the ‘second level’ of the UOS™ provides higher level, more aggregated intelligence from individual building scale to an entire city or a fleet of distributed devices. This layer collects, manages and provides insight to data, ensures that data is propagated quickly to where it is needed and provides an Application Program Interface (API) for applications to leverage.

**Applications (PlaceApps)** – again these are enabled by the UOS™ and may run in a context that is fully provided by the UOS™ and its platform, but should not be considered part of it (although certain applications may well largely consist almost entirely of exploitation of the services that form the UOS™ API). Applications enable users to interact with data and controllable equipment in the urban environment, subject to access granted by the security model. Applications are portable across UOS™ deployments, opening up a mass market for smart urban applications because the details of equipment interfaces are abstracted from the application by the UOS™ while the UOS™ is always the same consistent platform.
The Benefits of the UOS™ Architecture

The UOS™ ensures scalability and reliability principally by careful control of distribution of functionality. In general the UOS™ seeks to centralize management but distribute execution.

The key differentiators of the UOS™ architecture over traditional Building Management solutions are as follows:

- A unified sensor network provides cost benefits of an ‘all-shared’ sensor and actuator infrastructure, as well as richer insights through ‘naturally aligned’ data
- Converged networks provide efficiency and better performance with contained cost
- Lower cost deployment through the use of commodity hardware and software
- Highly redundant architecture eliminates single points of failure
- Race-proven controls architecture and highly reliable network hardware helps ensure primary reliability of control functions
- Control algorithms are centrally deployed and can be updated in real time
- UOS™ scales to larger control domains through use of near-real time data propagation
- Historical data allows for predictive models and continuous optimization of performance
- Abstraction models for capital equipment enable ‘pre-integration’ of hardware, substantially reducing commissioning cost and risk while enabling more comprehensive control
- Consistent application API provides a mass market for applications, stimulating the ecosystem that builds them and commoditizing the availability of functionality for UOS™-equipped developments

The UOS™ in Practice

UOS™ Sensor Network

A set of sensors in a given room in a building constantly detect conditions and relay this information through the sensor network to the controls level of the UOS™. The controls level knows either through supported protocols or a driver application how to correctly interpret the data coming from the sensors.

Control Application

This information may well be used in the first instance by a number of Control Applications resident in the UOS™ RTC™ core which manages and supports these applications. For example an HVAC application will take into account factors such as temperature, humidity, air quality, weather conditions and occupancy, predicted or advised arrival / departure times of occupants, and energy cost and availability to determine whether adjustments to the current conditions inside the room are necessary.
Other Information

Some of this information is obtained from local sensors, other information comes from elsewhere in the urban environment (or outside) and is relayed via the UOS™ Supervisory Level’s Urban Service Bus. The application when active runs continuously, taking account of changing parameters to determine – for example – the optimal position of a motorized flap which increases the amount of heated / cooled air ducted into the room – and/or the velocity of any airflow-boosting fans fitted.

Algorithms

Mathematical algorithms for this control constitute these control algorithms, which are usually built as the building is designed alongside choices about building physics and materials used. These algorithms can then be continually fine-tuned via the UOS™ (see below).

Supervisory Level Feed

In addition to these immediate control functions, information is also continually relayed to the supervisory layer of the UOS™. This achieves two purposes:

- The historical database is continually updated with what is being measured within the room and the entire development;
- Information required by other subsystems is relayed in near-real-time to those systems

History = Prediction

The collection of history data allows for very precise forecasting and prediction to be achieved. For example, being able to predict when an owner is likely to reach home or what settings they are likely to want for their climate control given outside weather conditions and previously observed patterns.

Urban Service Bus

The relaying of information to other subsystems is achieved using the Urban Service Bus. This employs a ‘publish-subscribe’ paradigm to allow other control applications / large scale control applications to subscribe to information available in the system as a whole. For example, it’s likely that the control application which maintains the temperature and airflow in the ducts of the HVAC system is physically separate from the one maintaining our room’s air flap. But this application appreciates knowing when the airflap is moved for all rooms connected to that duct, as it maintains a model of demand on the system so it can continually adjust its output to suit.

Applications

Our simple story doesn’t end there. Let’s say the apartment owner gets home and wants to make a change to his HVAC settings. His intended setting for this room is a variable held somewhere in the UOS™ and made available to the control application. But how can he change it?
PlaceApps
This is where PlaceApps come in. PlaceApps – the Living PlanIT term for user-facing applications in the urban environment – allow people to interact with systems. A PlaceApp for HVAC might run on a permanently mounted wall panel which looks like a traditional thermostat, and on a smartphone application, and on a TV screen... etcetera. The PlaceApps are built using Service Oriented Architecture – where much of the application content consists of services that are bound to simple UI elements, making it easier to write applications that run well on multiple devices. So via some PlaceApp the user makes a change, which is fed in near-real time to the control application which starts to make adjustments. But the fact that the user made the adjustment at this time and from what location is also recorded so that this can be used later on to better predict the users requirements.

Portability
Furthermore, applications correctly written for one UOS™ deployment will work in other UOS™-equipped developments, because the platform is consistent and the details of interfacing with specific types of equipment are abstracted from the application by the UOS™. This opens up a mass market for applications as opposed to their development inherently being bespoke and less efficient.

Optimization
But even now there is a coda to our story. All of the algorithms that run the HVAC system together with every algorithm in the urban environment can be continually fine-tuned to perform more efficiently under real-world conditions. What-if scenarios and optimization techniques using artificial intelligence can be played against the history data collected. Once algorithms are found which perform better than the initial algorithms, they can be deployed without downtime thanks to update facilities built into the UOS™ RTC™ core. This means that Living PlanIT equipped buildings get more efficient in the first years of their life, not less. This continuous optimization of all urban functions drives efficiency, and reduces the environmental impact of buildings and what people do in them.

The UOS™ and the Distributed Cloud Architecture

Controls Layer Distribution
The UOS™ integrates the controls layer with the network infrastructure to enable the running of the most time-critical functions at the edge of the network, minimizing latency and ensuring performance. This provides for the isolation of ‘chatty’ information from the rest of the network (for example, “don’t send new history data until a sensor value changes”). The UOS™ also runs within the security context of a local subnet, which forms an important part of the security framework.
This distribution also makes it relatively simple in most environments to provide redundant infrastructure, something unknown in the building controls market. For example a network device that provides control functions for one floor of a building can now be cross-coupled to another floor, so that if one controller fails the other can take over. The architecture is also inherently resilient in that control applications can function autonomously to maintain mission critical functions, even if isolated from the rest of the UOS™ network. Local storage allows buffering of historical data for many hours before any information is lost.

**Supervisory Layer Distribution**

In an urban built environment, the supervisory system is also distributed. In general the UOS™ is distributed in ‘clusters’ around a development or city/region, rather than centralized in a data center. The clusters will usually be located in an equipment room in a multi-tenant building and can be integrated with the building HVAC / district heating/cooling system for maximum thermal efficiency and minimum investment cost.

This approach has several advantages. The clusters will operate autonomously – given power – if disconnected from the rest of the UOS™ / urban network. Equally UOS™ clusters will usually be deployed with spare capacity, to be able to take on the most mission critical functions from another cluster if it fails or needs to be brought down for maintenance. When not used for redundancy, this spare capacity can be used for ’roving’ loads such as analytics and simulation, or sold as compute capacity to office tenants or other partners, or throttled down to eliminate power loss.

Data redundancy is also achieved more elegantly than in the traditional ‘replicate the data center’ model. One copy of data is always stored in the local cluster, because this is where it was generated and will most commonly be referred to again. Equally at least one copy of data is stored in a location remote from the originating cluster. Finally, aggregated information is stored in synchronized ‘supernode’ clusters – larger than average clusters with more of the historical city data stored in them - in order to optimize performance of the analytics engine.

The least compressed, most ‘raw’ data – especially video imagery – is typically kept for forensic purposes for a few days only; it is usually kept local to the cluster, minimizing network load.

All UOS™ functions are designed to enable this ‘multinode’ deployment model and optimize the benefits presented by this physical architecture, while mitigating the disadvantages (such as avoiding database joins that are propagated over multiple dispersed systems). As functions can be redistributed across the architecture at will, this is considered to represent a ‘distributed cloud’ model as the principal characteristic of location independence still applies.
Supervisory Layer – Tiering

As has been noted above, the UOS™ scales out by way of a distributed multimode model. Further tiers can also be added in a ‘vertical’ domain, aggregating functionality in a ‘pyramid’ topology. For additional optimization, the routing of information from one node to another does not need to follow purely hierarchical lines – the routing model can be continually tuned to meet the requirements of the specific deployment and applications in use.

Supervisory Layer – Cloud Deployment

Certain solutions lend themselves better to a more classical cloud deployment where functions may well be deployed in a typical data center hosting environment. The UOS™ also adapts to this paradigm, with the ability to scale up and out still being critical to the way in which functionality is deployed on multiple virtual machines to meet demand. These solutions are likely to be used most commonly in M2M solutions, and where the ability to carry a distributed local infrastructure is minimal – for example when integrating the UOS™ with streetlights, or when retrofitting single family homes.

Leading ideas for a sustainable urban future

Deployed in association with an extensive partner and stakeholder ecosystem, developers, building owners and service providers use the UOS™ to envisage, design, manufacture, assemble, operate, service, maintain and decommission buildings more efficiently, improving performance in terms of environmental, economic, and social sustainability.

The smart infrastructure deployed in these developments – built around the Urban Operating System or UOS™ - provides a platform for both economic growth and for adding capabilities to these cities and urban communities on an ongoing basis through the development of value-add applications.

The data collected also enables a new style of integrated reporting for cities in which all stakeholders can receive accurate information on financial and nonfinancial (environmental, social, and governance) performance and the relationships between them. Through integrated reporting and PlaceApps using this information, the city and its citizens can make better decisions about their use of financial, natural and human resources, thereby creating a more sustainable city. In addition this type of reporting and its associated PlaceApps will also dramatically improve the level of engagement a city has with its citizens.
Interfaces and Protocols

Overview

The Living PlanIT Urban Operating System (UOS™) is intended to provide the platform for smart city development and deployment and leverage M2M-capable devices. As such, it is critically important to strongly interoperate with a wide range of devices, protocols and other platforms. The UOS™ has been designed to ‘rationally maximize’ this through the following principles:

- Completely open, platform-neutral access to all core UOS™ interfaces
- No ‘private interfaces’ accessible only to Living PlanIT or ‘favored’ partners/customers
- Support for key industry standard protocols where these have significant traction/take-up and do not overly bias the platform towards any particular subsector of the market
- Use of de jure ratified industry standard protocols and interfaces where possible and practical
- Use of de-facto industry standards where de jure standards do not exist, and where possible and practical
- Use of Living PlanIT-created protocols where effective standards do not exist; make these openly published and available to the market; support adoption by standards bodies when appropriate (based on traction)
- While adhering to the above, avoid the pitfall of trying to be a ‘universal adaptor’ in order to ensure lightweight footprint and strong performance at scale

This document describes each of the interfaces that are expressed by the UOS™ and the current plan for the architecture of these interfaces and what protocols/data formats are supported. It should be noted that not all features listed may be available in all versions of the product, and may not yet be supported in current available versions. The final section of the document contains further details on the Living PlanIT philosophy with regard to open-specification or open-protocol solutions and how these differ from the commonly-requested open-source approach.

Architecture

In order to comprehend the role and rationale for UOS™ interfaces, some understanding of the UOS™ architecture as explained and depicted earlier in this document is necessary.
UOS™ Application Program Interface (API)

The UOS™ API is designed to be the most commonly used interface, as it is fully expected that more entities will write applications for the platform than will integrate existing platforms or data feeds, or integrate sensors or assets requiring actuation. The API consists entirely of web services delivered over IP-based networks, and may reasonably be described as a pre-built Services Oriented Architecture (SOA). Living PlanIT recommends use of SOA architectures in designing and deploying applications on the UOS™.

UOS™ uses the Open Data Protocol or odata as its query mechanism for data retrieval and the intrinsic filtering of subscribed data. This is implemented on top of the UOS™ Data Model, which is described in the UOS™ Data Model Specification and the UOS™ Software Development Kit (SDK).

Currently all UOS™ API calls are expressed as RESTian web services, with both XML and JSON available. The UOS™ currently provides support for http and https. In most cases https is strongly preferred but there are scenarios where http may be viable and the penalty cost of https need not be paid. Otherwise, support to http calls will often need to be disabled / blocked in a production context. See UOS™ Security Model documentation for more details.

Support for SOAP and WS-I profiled services will be provided in Q4, 2012.

The UOS™ API is divided into 3 functional classes:

1. Core APIs – these provide access to the main UOS™ Core services including data access, message bus, publish-subscribe system, built-in analytics and simulation support, security and identity model, and management and monitoring functions.
2. Application Support APIs – these provide services that facilitate the hosting of applications directly on the UOS™ Compute Platform. If these services did not exist, applications with server content leveraging the UOS™ would always require at least two platforms – the UOS™ instance, and a separate hosting environment. While this deployment model is always inherently supported, in many cases it will be more convenient to host on the UOS™ platform. Example APIs include notification services, application-local persistence, and application service deployment. It should be noted that delivery of class 2 APIs generally lags classes 1 and 3 because it is not strictly necessary for UOS™ deployment or usage.
3. Functional APIs – these abstract and normalize capabilities provided by actuable capital equipment interfaced with the UOS™. For example, multiple families of luminaire/controller from multiple lighting providers can be controlled by the same API without the application developer ever needing to care what type of equipment he is controlling.

The UOS™ API is currently available to Living PlanIT partners as part of the SDK. Broader availability is planned and will be announced in due course.
UOS™ DSI (Distributed Scale Interface)

The UOS™ DSI (Distributed Scale Interface) supports the interaction of the two core UOS™ layers – the controls and supervisory layers – and also enables the distribution of functionality across multiple instances of UOS™ Core, providing extensible flexibility in how the UOS™ is deployed and the ability of UOS™ to scale to extremely large scope. For further information, consult the UOS™ Architecture Guide.

While this interface is conceived principally as an ‘internal’ interface to support the operation of the UOS™ it is not private. Opening up this interface brings the following benefits:

- Flexibility of deployment especially in M2M scenarios – devices may not need the services of the controls layer and may be able to interface with the supervisory layer directly
- Partners and integrators can use the UOS™ Supervisory layer with legacy controllers or their own DSI-compliant controls architecture
- Partners and integrators can use the UOS™ Controls layer with no Supervisory layer or with an alternative Supervisory layer (not recommended, but feasible)

Both the UOS™ Controls layer and Supervisory layer require the DSI to be networked using IP, although the use of bridging technologies in-between that is sufficiently performant and transparent cannot be ruled out.

The DSI has two transport solutions implemented which are used for differing purposes. Generally both need to be implemented for full DSI support. These are:

- RESTian web services interface (http/s): used for low volume sensor data, high priority traffic from the controls layer, and for command and control purposes.
- Named pipe interface (TCP/IP): used for message bus / high volume sensor data / bulk movement of time series information.

For the Supervisory layer, the RESTian interface supports an API to send sensor data to the UOS™ using the Living PlanIT TSP (Trivial Sensor Protocol) format. Additional calls include calls used in coordinating UOS™ Supervisory layer multi-node behavior and the management and monitoring interfaces for UOS™ Core.

For the Controls layer, the RESTian interface supports an API to send control instructions from the UOS™ API to controllable assets connected to an instance of the UOS™ RTC™ (controls layer host). Additional calls include the configuration of the UOS™ helper application and the RTC™ piece of the management and monitoring interface is also implemented.

The named pipe interface supports the high speed message bus used to transport data matching the pub-sub mechanism and also the bulk sending of sensor data using the Living PlanIT SDP (Sensor Data Protocol) format.
For inter-layer and inter-node use Living PlanIT provides two optimized data formats which correspond to the UOS™ Data Model. The Trivial Sensor Protocol (TSP) is intended to make integration with UOS™ very simple and is optimized for low frequency data updates. This format is already published by Living PlanIT and is available under a free license.

For efficient high bandwidth communications of multiple sensor streams, the Sensor Data Protocol (SDP) should be used. This format uses multiple forms of compression and can handle significant variance in multiple concatenated streams optimized for hundreds of individual sensors per stream. SDP is out of necessity more complex than TSP but will also be published by Living PlanIT under a free license in order to facilitate its use in aggregators or by 3rd party controller manufacturers.

**UOS™ SCI (Sensor & Controls Interface)**

The UOS™ SCI provides the means by which sensors and actuators are integrated with the UOS™. Integration occurs in three ways:

- The use of a standard protocol
- The use of an existing ‘driver application’
- The development of a new driver application

In addition the SCI will also provide the point of control for downstream devices (aggregators, smart sensor boards) which provide UOS™-supported management, monitoring, and security interfaces. This will be described in a separate document.

This interface is designed to use IP-based networking. Other types of network must be adapted to IP using downstream gateway devices (with the exception of Serial Port I/O which may be handled natively by some RTC™ hosts).

**Standard Protocols**

Standard protocols supported include the following:

- Living PlanIT Sensor Data Protocol (SDP) and Trivial Sensor Protocol (TSP)
- Modbus (RTU and TCP)
- OPC (DA / AE)
- X10
- TAPI
- SMTP
- SNMP
- SNPP
- DDE

Others will be added based on market penetration and partner request.
Adaptable Transports

The 'native' transport for the SCI is IP-based networking (mostly TCP/IP, although UDP/IP is also supported). All current Living PlanIT-supported environments handle both IPv6 and IPv4 depending on customer requirements, with IPv6 being recommended in order to avoid address availability issues.

In addition the aggregator layer allows for other networks and transports to be adapted to IP for use with the UOS™. A non-exhaustive list of such transports follows:

- Current-based sensing devices (via A-D conversion)
- Serial Port
- USB
- Zigbee, 6LoWPan, and other 802.15.4-based networks
- Z-Wave, EnOcean and other non-802.15.4 mesh networks
- Wi-Fi (802.11x)
- Wireless Cellular Networks (3G/4G/+)
- Modbus (serial forms)
- BACnet
- LonWorks

Existing Drivers / Legacy Systems

These drivers are developed by Living PlanIT and its partners as new equipment is interfaced with the UOS™. New drivers are being continually certified and made available. Below is a sample of existing certified drivers (non-exhaustive list):

- Lighting Systems:
  - Philips Dynalite
  - Philips ColorKinetics
  - Philips Outdoor
- Fire Panels:
  - Andriod
  - Detectomat
  - Eltek (eComm protocol)
  - Esser 5008, 8000
  - Gent 3400
  - Hochiki
  - Howfire
  - Kentec
  - Kidde Vega, Proceyon
  - Multi Alarm IFAX
  - Notifier ID50, ID1000, Morley, AFP4000
- Protec AN95, 5400, 6400
- Schrack Integral N3 and Maxima
- Siemens Cerberus Algorex, CZ10
- Thorn Minerva
- Trident
- Wormold

- PA, Audio, and TAPI systems:
  - Audionics Audio Router
  - Cisco Call Manager
  - Complus Teltronic Intercom GE100, GE200, GE700
  - ComTalk PA system
  - Ericsson MD110 (TAPI)
  - Mitel 33
  - Vortex VX

- CCTV systems:
  - American Dynamics Micro Power EP Switching Matrix
  - American Dynamics Intellex
  - Avigilon
  - Axis encoders, decoders and IP cameras
  - Barco Hydra video display
  - Baxall 7000 matrix, ZMX, ZT6 matrix, Pyramid
  - BDL UVSS
  - Burle/Philips/Bosch Allegiant LTC8000 series matrix
  - City Sync ANPR
  - COE telecommand
  - Dahua DVR
  - Dallmeier encoders, decoders and DVRs
  - Dedicated Micros DVST, Uniplex Series 2, Unicord
  - Drax 17 Duplex Multiplexer
  - Ernitec Matrix Series (ELP)
  - Geutebruck VICros III matrix
  - Hikvision DVRs
  - Honeywell/Maxpro MAX-1000
  - Ipsotek IP Series IPS: IPS-4, IPS-10, IPS-16
  - ISolutions iRecord 5000 video system
  - JVC SR-S970EK VCR
  - March Networks
  - Meyertech ZoneVu II video matrix
  - Mitsubishi HS-S5600(BRS) VCR
  - Molynx Visilynx matrix, TTX309 interface to 600 series matrix
  - NiceVision video system
  - Panasonic AG-7630, AG-6040, AG-6760 VCRs, WJ-SX550 Matrix
- Pelco Endura or DX8x00 via X-Portal XMI
- Philips matrix
- Photon Surcha telemetry receiver
- Photoscan dimension matrix
- Plettac POSA video system
- Plettac VAZ300 matrix
- Synectics X250M series matrix
- Vicon Kollector on Vicon NET (limited functionality)
- Vicon Nova 1500 matrix system, VPS series
- Vision Factory Montage Multiplexer
- Vision Systems Adpro VST10 Fastscan

- **BMS/BEMS controllers:**
  - Allen Martin energy management system ACM
  - American Automatrix Public Host Protocol (PHP)
  - Andover Infinity, CX and CMX controllers via CLS
  - Andover Netmaster and AC8 controllers
  - AutoMeter IC500, IC700 and C-Matic 3300 ACM
  - Cisco Network Building Mediator (Richards Zeta)
  - Johnson Controls IU-9000 interface to N2 network
  - Leibert 9000 UPS
  - Manas energy meters database
  - Mitsubishi FX series PLCs
  - Next NCS2000
  - Nudam ND-6050, ND-6053, ND-60xx
  - Optimal energy and SCADA database
  - Polaron Light link 98 system
  - SAIA PLCs via S BUS protocol
  - Sauter printer port
  - Staefa Nico PLC
  - Stark EMS
  - Tour and Anderson Sys7
  - Trend Control Systems Trend IQ

- **Security systems:**
  - BDL UVSS
  - Bold Communications RX2000 Alarm Receiver
  - BT Redcare
  - Casi Rusco Picture Perfect access control system
  - City Sync ANPR
  - DSS200 Nurse Call System
  - Europlex intruder panels
  - Fermax door entry system
  - Honeywell Galaxy, Series 3, Classic, Dimension
• Marconi SMS
• Menvier intruder panel
• Morse Watchmans Keywatcher key management system
• Northern WinPak access control system
• Scancom Scantronic intruder panels
• Shop Alert system
• Smiths Detection Explosive Particle Detection system (EPD)
• Software House C CURE 800 access control system
• Solid Card RiTA access control system
• TDSI Ultra Guard ACU (events only)

• Messaging / Paging systems:
  • Multi Tone Multi Tone Access 300 MIT system
  • Nevotek Nevotek IPTV system
  • Nokia Nokia 30 GSM connectivity terminal
  • Scope Scope paging system
  • TextForce Text force SMS service

• People Counting Systems:
  • Experian Footfall
  • Crowd Vision

• Car Park systems:
  • Dambach PGS
  • Erben Control Systems Database

• I/O modules:
  • Advantech Data Acquisition Modules
  • Audon Electronics SOM1 Serial Output Module, SPO-RL8 4 Relay Output Module

New Drivers

Living PlanIT partners can develop their own driver applications using the Living PlanIT SDK and appropriate tooling. Developing a custom driver allows for any arbitrary ID convention, multi-sensor concatenation, data format or variance to be supported for sensors, and is typically required for actuators in order to expose functionality provided by the equipment (although a general purpose actuator driver is in development).

The Living PlanIT UOS™ SDK provides more information on how to develop a custom driver. A supported development tool is required, at present this is limited to Mathworks MATLAB/Simulink. Alternative tools will be released at a later date.
UOS™ EAI (External Asset Interface)

The UOS™ EAI provides a generalized integration server capability in order to support the integration and alignment of external data sources and complex integration with other external assets such as ERP and billing solutions.

The integration of external data sources can in many cases be accomplished using a native UOS™ interface which supports XML-based data feeds with XSLT transformation and the ability to embed customized functions. This interface is described in the UOS™ SDK.

More complex integration scenarios are supported using the optional Microsoft BizTalk Server component which can be deployed in selected UOS™ Supervisory layer nodes. This allows all of the adapters and sophisticated messaging and orchestration functionality supported by BizTalk Server to be used to adapt external functionality to the UOS™ Message Bus, Data Model, Data Store and API. For more details please see the ‘Using BizTalk Server to extend UOS™ Interoperability’ document.

Further extended interoperability options may be available in future releases in scenarios where the UOS™ is deployed alongside other smart city solutions which are focused on the application part of the stack. As an alternative, customers can integrate their own high-level, low-speed generalized service bus with the Living PlanIT Urban Service Bus by making use of our core API.

Open Specification Philosophy

Innovation and Open Standards

Innovation in the technology sector generally consists of leaders developing from time to time new concepts which are implemented in software, firmware and hardware that significantly advance an area or areas of technologies. These innovations can through mass adoption become de facto standards and many ultimately are at least in part canonicalized as de jure standards through recognized international standards bodies. Typically the lag time from proprietary implementation through to agreement on an international standard is many years with 5-10 years not being uncommon once implementation patterns are established for a vertical or sector.

For customers to reap the benefit of these technologies during this period requires careful selection of a vendor who is pragmatic about leading, supporting, and adopting standards. Excess effort in trying to provide a ‘universal adaptor’ or participate too early in too many standards activities – which may be competitive – usually results in a lack of effective implementation and velocity, to the customer’s detriment. Equally a hidebound or extremely proprietary mindset may provide short term benefit for long term pain as the cost of inflexibility is brought to bear.

Living PlanIT is committed to support:
• Key standards that are mature, stable, and adopted by relevant industry sectors.
• Adoption of Living PlanIT technologies and protocols by other industry participants and ultimately standards bodies where sought.
• Emerging standards that have significant industry traction and are appropriately backed by Living PlanIT partners.
• An ongoing effort with our partners to continually monitor and participate appropriately in the pragmatic development of relevant standards.

In so doing it is Living PlanIT’s stated intent to provide customers with solutions that will work and interoperate strongly now and in the future and as such protect the value of investments made by customers and partners today – in other words to futureproof UOS™ implementations.

Open Source vs. Open Specification / Open Protocol

A frequently asked question is whether Living PlanIT’s Urban Operating System (UOS™) will be open source, the implication being that this would be a good thing from a perspective of future-proofing the application of this technology.

Living PlanIT has no intention to make the UOS™ open source. Instead, we are adopting a policy of open specification or open protocols, where every interface to the UOS™ - as described above in this paper - is fully documented and freely licensed to new and existing partners, enabling a vibrant ecosystem to be forged which inter-operates seamlessly with the UOS™.

To make the UOS™ open source would mean the following:

• All UOS™ source code would be in the public domain
• Other players would have the right to contribute to the code base
• No one entity would be in control of the direction taken by the UOS™

We believe this would manifestly not be in the interests of the community served by the UOS™ for the following reasons:

• Open source has been shown to provide value in some cases for a subset of a technology marketplace that is commoditized, and providing stability and evolutionary improvement is more important than innovation (examples include Linux, Apache). There has not been an example to date of a large scale successful open source project in an area of rapidly developing technology. The lack of coordination and the absence of focus and investment which a commercial venture can provide means that this approach will struggle to provide the rate of development needed to make such a platform a success.
• Conversely, in order to justify the investments that it will make in this platform, Living PlanIT needs to be able to monetize the development of the UOS™. This would be very difficult to do in an open source environment, as we would enable the rapid development of our own competition.
• The UOS™ needs to be highly scalable (both up and down), highly performant, efficient in everything it does, and uniquely run on a highly distributed private cloud architecture.
This requires very precise and well thought through architecture and design with extremely holistic integration of different subsystems. This would be almost impossible to achieve in an open source project with multiple contributors, even if everyone was diligently working towards positive outcomes.

- The UOS™ needs to be particularly secure in order to be successful as the wide scope of information captured and controls made available makes the platform a target for attack. Opening up the source code provides ample opportunity for malfeasants to inspect it to look for attack pathways and removes the potential for undisclosed security traps. This is clearly undesirable.

- Since the UOS™ will collect more data which directly informs or can be used to derive information about the activity and behavior of residents and visitors, a highly robust identity and privacy metasystem is essential, and has been designed into the UOS™. Making the inner workings of these mechanisms open source is counter to the design objective, as it would make it far easier for someone to inappropriately access data that they have no rights to.

- Open source projects are typically dependent on the efforts of a community for support – professional support is only available through commercial ventures built on the back of some version (or distribution) of an open source project (e.g. Linux distributions). Costing studies show that the lifetime cost of such solutions usually is comparable with the lifetime cost of commercial software – in other words no cost saving is achieved. Additionally the services and support available usually fall short of those available from commercial solution providers. Since developers using the UOS™ will be looking for high degrees of operational support for an extended lifetime, a well-founded commercial venture is a better platform on which to build such support.

- Building extensive open source solutions on top of proprietary platforms is rarely done, because this somewhat obviates the rationale for making the solutions open source in the first instance. As a result, underlying platforms tend to be open source also. This reduces choice for the solution developer, and may mean building suboptimal solutions compared to existing commercial offerings particularly in terms of the disadvantages already noted above.

- Broad adoption of an open source platform tends to lead to a greater degree of divergence and subvariants. And yet for a vibrant ecosystem and maximum choice in interoperable solutions and components, broad adoption is desirable. Since the interfaces for every instance of the UOS™ are the same, equipment integration or applications developed for one project are portable to another, creating a mass market for our ecosystem. Platform divergence would damage this critical commercial dynamic.

- As demonstrated by the growing number of patent wars between industry players, trolls and other contributors, it becomes increasingly hard to warranty that in the development of open source based technologies that at some point intellectual property from 3rd parties has not been unknowingly or deliberately incorporated into the body of the source.
• Where such impingements on 3rd party IP are found, networks or cloud based solutions (as just two examples) can find themselves under judicial injunction and in default of significant damages for loss of revenues for the originator of the intellectual property.
• Some open source licenses exert strong conditions on the developer of works that include the open source code, up to and including the need for such derived works also to be open source. This can cause significant unintended diminution and leakage of IP.
• As a result, Living PlanIT develops and delivers all products free of open source code – furthermore it warrants to our customers and partners that no such code has been incorporated in our platforms and indemnifies them from all such claims by 3rd parties. It should be noted that our supervisory layer runs exclusively on a Microsoft platform – and Microsoft extends similar warranties to its customers and partners building on that platform. By comparison, this is not available from Linux distributors.

In comparison the benefits of our open specification approach are:

• The core of the system is produced by a single commercial entity, leveraging the best available technologies from the marketplace.
• This core can be optimized from concept through to deployment, ensuring required levels of scalability, performance, and efficiency.
• This core can also be secured and designed to ensure that constituent data is appropriately protected.
• The core can be supported through support models synergized with core solution development.
• Broad adoption means that more investment can be made in the common core platform to the benefit of the entire community.
• The availability of all interfaces through open specification means that partners have the complete access to the system they require to get the most out of their integration with it and it opens the door to any partner who wants to build a solution, without having to make any changes to the core system – which therefore does not have to be rebuilt to accommodate new partners or content.
• Keeping these interfaces stable and functional – best achieved via a commercial organization - maximizes value to the community.
• Open specifications tend to grow vibrant, commercially viable ecosystems of expertise (eg. OPC, Wi-Fi, WS-I).
• Open specifications reaching critical mass provide the best platform on which to build industry standards as this accelerates the standard reaching the market in usable form and therefore adoption.

It should be noted that Living PlanIT may be prepared to put certain items of non-core code into the public domain, or even release as open source under appropriate licenses that do not infringe on customer / partner IP. This may non-exhaustively include sample code for device
integration and applications, and firmware associated with reference hardware designs. The decision to do this will be a strictly commercial one taken by Living PlanIT only.

As adoption of the UOS™ platform increases and the partner ecosystem expands, Living PlanIT is prepared to consider the following to ensure that the community is optimally protected and supported:

1. Take open specifications used for UOS™ interfaces to ISO-affiliated standards bodies to convert a de facto standard into a de jure one;
2. Place critical UOS™ source code into Escrow for the benefit of the partner community in the event that Living PlanIT, its owners, and/or successor companies ceases to be able to support the UOS™.

We believe these measures achieve all that is looked for with an open source solution, without any of the manifold disadvantages.

The roadmap for Living PlanIT making its specifications broadly available as part of the open specification / open protocol approach is documented in this paper, which will be updated as the platform and the strategy continues to develop. For the latest information please see our corporate website at http://www.living-planit.com.

The open interfaces and multiple protocols supported by the UOS™ provide a wide and efficient set of paths for integrating heterogeneous scenarios with the UOS™. These will continue to evolve as standards develop and emerge in order to support the most efficacious and broadly supported interoperability options. If further information is required please refer to the documents referenced herein or alternatively contact Living PlanIT.