SFpark
Technical Manual

Guide to the information technology behind SF*park*

SFpark | SFMTA Municipal Transportation Agency

June 2014

SF*park*

Technical Manual

Guide to the information technology behind SF*park*

SFMTA
Municipal Transportation Agency

Acknowledgements

Thank you

This document summarizes the technical aspects of the SF*park* pilot project, and in particular the approach to data management. This foundation for the SF*park* project represents a significant achievement, and has received numerous awards, including:

- 2011 SF Weekly Web Award—Best Local Government Site
- 2011 Department of Defense Service Oriented Architecture & Semantic Technology Symposium “Best of Show” Awards
- 2012 Excellence.gov Awards—Excellence in Innovation: Mobility
- 2012 7x7’s Best of San Francisco for ‘Real world technology innovation’ category
- 2013 Sustainia100 Top 10 Innovations in “Cities” category
- 2013 Harvard Kennedy School’s Top 25 Innovations in Government
- 2014 Oracle Spatial and Graph Excellence Award, innovator category
- 2014 Presenter at Business Intelligence, Data Warehousing and Advanced Analytics Summit for innovative use of BI, DW, and GIS technologies to advance smart cities

Launching the technology to implement and evaluate the SF*park* pilot project depended on the dedication of a multidisciplinary team within the San Francisco Municipal Transportation Agency (SFMTA) as well as strong partners.

The planning and development of the SF*park* system would not have been possible without the knowledge, expertise, passion, and dedication of many people, including:

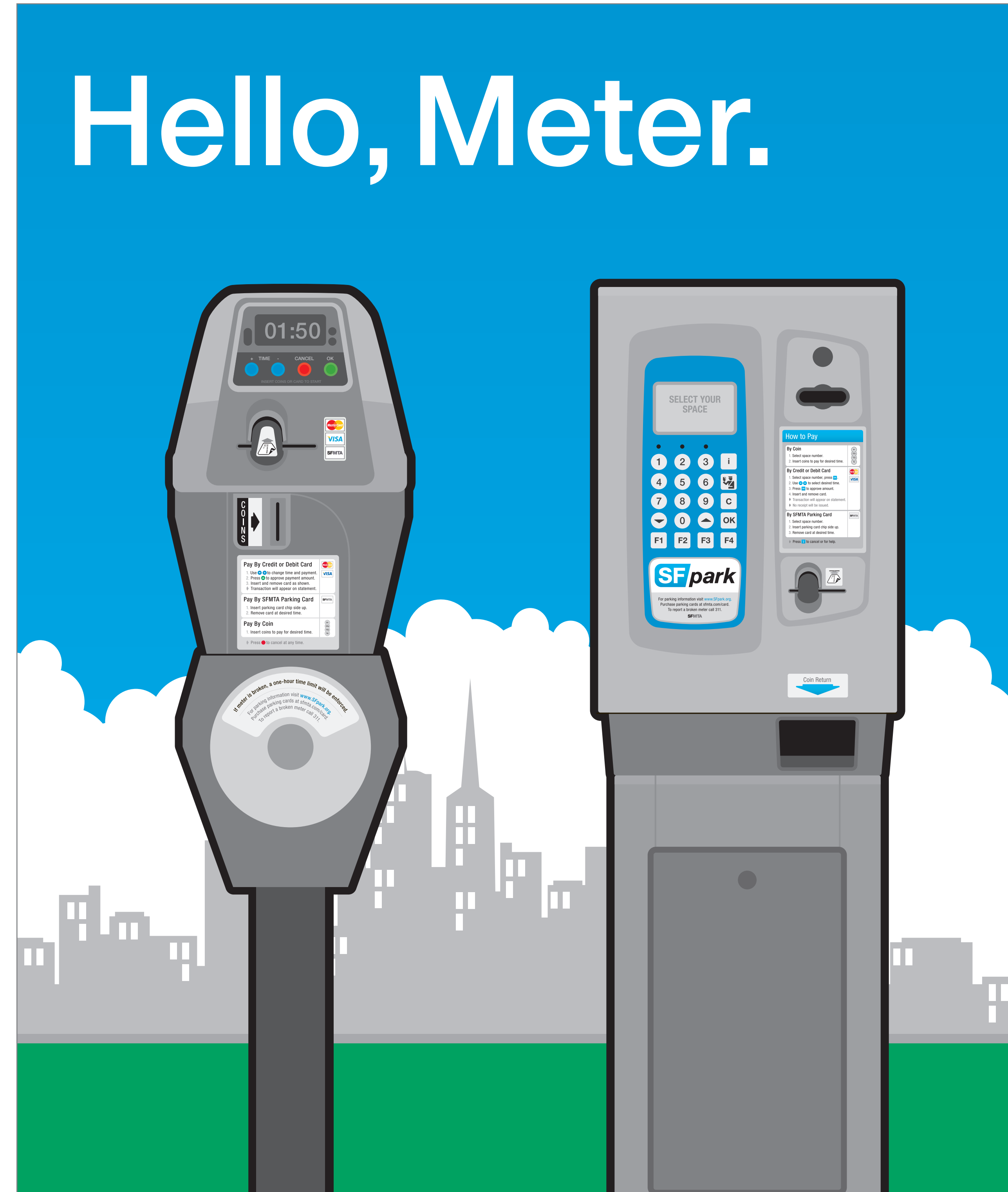
- Jaime Ballesteros
- Alex Demisch
- Donovan Corliss
- Eric Ganther
- Saunnie Hawkins
- Sanjay Kulshrestha
- Brendan Monaghan
- Steph A. Nelson
- Benjamin Perez-Goytia
- Mark Piller
- Mohammed Shaik Hussain Ali
- Ashish Shukla
- Randy Vakoc
- Sai Vakkalanka
- Eduard Zvenigorodskiy

The SFMTA would like to thank them for their efforts.

Organizational partners

The SF*park* projects would not have been possible without the generous financial support of the United States Department of Transportation (USDOT) and Federal Highway Administration as part of the Urban Partnership Program.

The production of this document was made possible by financial support by the United States Department of Transportation (USDOT) and Federal Highway Administration as part of the Value Pricing Pilot Program.



Contents

What's inside

1. INTRODUCTION

This chapter provides an overview of how the San Francisco Municipal Transportation Agency (SFMTA) developed the technical infrastructure of the SF*park* pilot project including goals, a system overview, and roles and responsibilities.

Purpose and scope

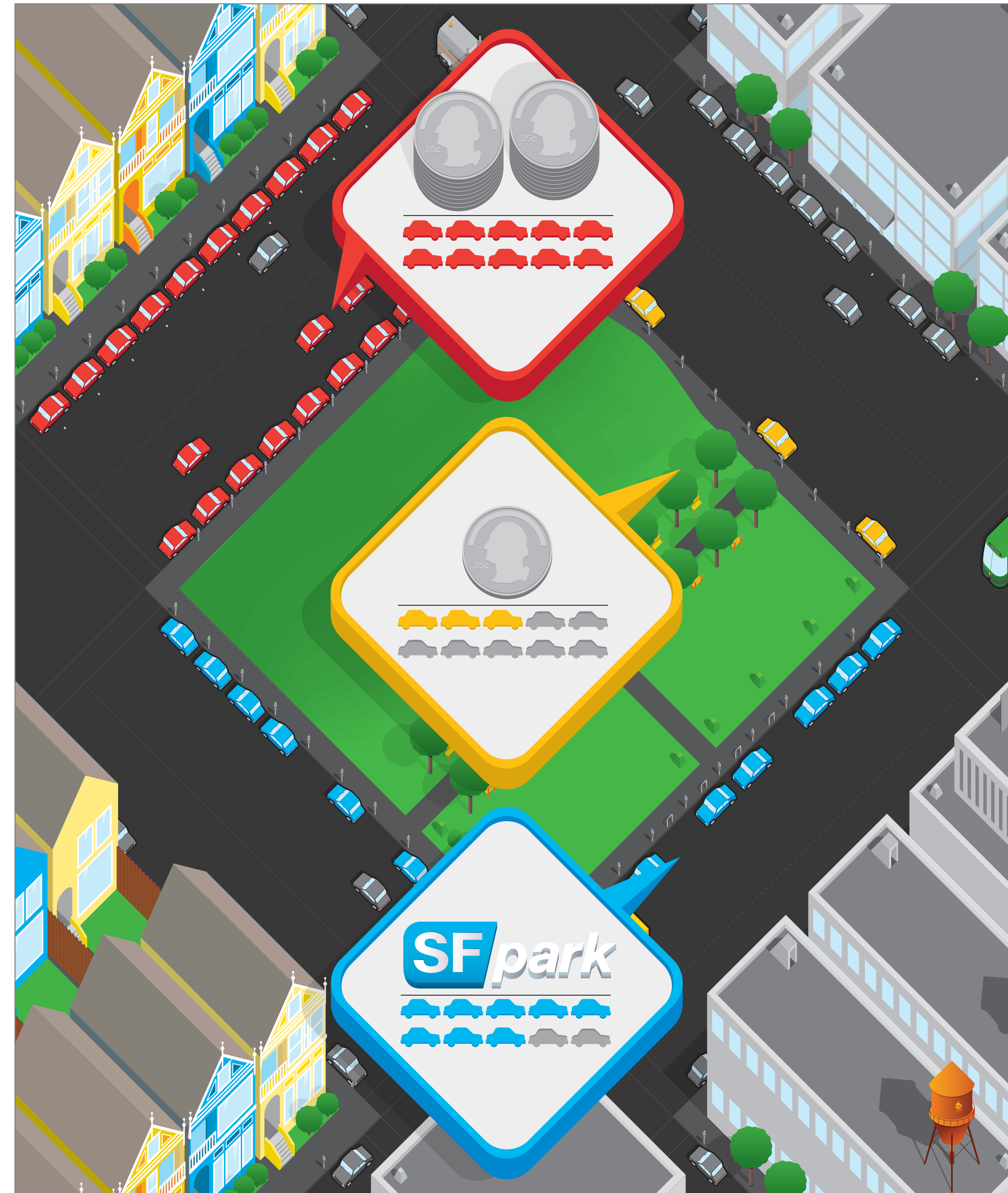
An overview of the purpose and contents of this document

The purpose of this document is to provide guidance for the technical groups involved with implementing SF*park*-like parking management technology. It documents the technical approach and high-level design of the data management system, and provides detailed reference material for the data management system the SFMTA developed for SF*park*. The aim is to provide a useful reference point for other cities that are considering relatively sophisticated parking pricing programs.

The SF*park* demonstration project has been a pioneering parking management effort. Because SF*park* represents the first time such a project has been undertaken, it had to overcome several novel policy, operational, and technological challenges that are a necessary part of parking pricing. Many cities ask the SFMTA for guidance as they consider implementing new approaches to parking pricing. As the SFMTA does not have the resources to counsel each city separately and adequately respond to their requests, this guide was meant to help provide general but detailed guidance in the technical area to help other cities learn from and improve upon what SFMTA did for SF*park* and, hopefully, save time and money.

The document describes specific technical choices of the SF*park* pilot project, attempting to create a technology-agnostic technical overview of SF*park*, including an overview of how the technical information technology (IT) aspects of SF*park* were implemented, issues encountered, and lessons learned. While not a formal technical manual, this document is intended to facilitate efforts in cities that choose to pursue the necessary IT improvements for more sophisticated parking management.

The intended audience of the SF*park* technical manual is the small but critical group of technically-inclined rather than policy-oriented staff at parking and/or transportation management organizations such as parking management system implementers, IT personnel (e.g., technical project managers, DBAs, system administrators), system integrators, and parking equipment vendors. As such, the document assumes a relatively high base of technical knowledge, but to assist more general readers acronyms are defined in the glossary.



SF*park* overview

A new approach to parking management

In 2011 to 2013, the SFMTA implemented SF*park*, a federally-funded demonstration of a new approach to parking management that combined improved information about parking availability and price, including real-time information from both on- and off-street parking spaces, and demand-responsive rate adjustments based on recent historical parking occupancy rates. Implementing this project required a suite of IT tools, many of which represent significant technological advances. To learn more about the scope of the SF*park* project, please consult the comprehensive overview found in "SF*park*: Putting Theory Into Practice."

Underlying all the analysis and operation of the SF*park* pilot project is an enormous amount of data management. Parking meters are now "smart" with a high degree of programmability as well as the ability to communicate wirelessly. Sensors in the roadway and in parking spaces collect data, and in the case of parking space sensors, communicate information in near real-time. All this data is collected, cleansed, and stored within the SF*park* system. Other processes combine this data with other data and restructure it to form a data warehouse. Access to this data warehouse is through a business intelligence (BI) portal.

The data warehouse enables the SFMTA to:

- Analyze parking occupancy so that SF*park* can make data-driven pricing decisions.
- Provide real-time parking availability information to the public.
- Operate and manage the city's on-street parking spaces.
- Monitor performance of meter, sensor, and garage vendors.
- Evaluate the SF*park* program as a whole.
- Conduct ad hoc analyses to support parking and transportation management decisions.

Parking sensors: Parking sensors are a nascent technology that provides data used in demand-responsive rate adjustments and real-time parking availability information. The federally-funded SF*park* pilot helped to catalyze this industry and was the first large-scale use of this technology, which surfaced many challenges related to installation and day-to-day operation. This technology will likely continue to evolve quickly in the years to come.

Real-time data: The SFMTA provides a real-time data feed (in JSON/REST format) with information about parking availability, prices, and regulations for both on- and off-street parking. The SFMTA uses this data feed for its own SF*park* app, and ensured that it is also available to other app developers and organizations so they could help to disseminate this information and maximize its social benefit. While the benefit of real-time data is hard to quantify, getting real-time data about availability and price to a wider audience meant that more people could make smarter travel decisions, thus maximizing its benefit.

Networked parking meters: The complexity of SF*park* pushed the envelope for parking meter management software. The SFMTA used lessons learned from the pilot to develop a new specification for a parking meter request for proposals (RFP) in 2013, which required parking meter vendors to enhance their products. This enhancement will enable the SFMTA and other cities to do more sophisticated and effective parking management and pricing in the future.

Roadway sensors: The SFMTA tested nascent roadway sensor technology to measure traffic volume and speed. A portion of the roadway sensors did not perform as expected, not transmitting data, doing so infrequently, or transmitting erroneous data points. Nonetheless, with transmissions every 15 minutes over two years and rigorous data cleaning, roadway sensors provided the SF*park* team with critical information for pilot project evaluation.

Data management tools: Part of the SF*park* project was the development of a set of complex and unique data management tools that enable project operation and evaluation. The tools represent a large investment that the SFMTA can leverage to increasingly use data to make more informed management decisions for not just parking and parking enforcement but also other areas of the transportation system such as transit. For parking management, these tools have already enabled much more sophisticated operations, contract management, reporting, evaluation, as well as to the ability to conduct demand-responsive rate adjustments using payment data rather than parking sensors to estimate historical parking occupancy.

Project references

Resources for further guidance and more detailed information

This section provides a bibliography of key project references and deliverables that have already been produced and are in the appendices. Refer to these documents for further clarification or specific direction on a topic or program area.

1. SF*park* post-launch implementation summary and lessons learned

This document summarizes SF*park* pilot project documents and lessons learned during project planning and implementation. Subjects include: parking management policy, administration & contract management, data collection & evaluation, parking technology, and communications. This web-optimized version includes links to a wide variety of project documents.

SFpark.org/resources/sfpark-post-launch-implementation-summary-and-lessons-learned-web/

2. Parking sensor performance standards

This clearly defines and outlines how the parking occupancy sensors operate and the methodology of validating data accuracy, including definitions of data accuracy.

SFpark.org/wp-content/uploads/2011/09/SFpark_SensorPerformance_v01.pdf

3. SF*park* availability service API reference

This documents the API information needed by end users and partners to effectively test retrieval and display of availability data from the SF*park* data warehouse. It is a living document, intended to be updated as required during the life of the project.

SFpark.org/wp-content/uploads/2013/12/SFpark_API_Dec2013.pdf

4. SF*park* meter vendor rate change protocol

This document outlines the protocol and provides a detailed outline of the steps involved in updating parking meter rates and displays via XML including timelines and roles and responsibilities.

SFpark.org/wp-content/uploads/2011/09/SFpark_RateAdjustment_v01.pdf

5. SFMTA on-street parking management policies

The on-street parking management policy document provides guidelines for future parking management changes. The goals are to communicate a simpler, transparent, data-driven approach for setting parking rates. Most importantly, it improves the transparency of how the SFMTA makes parking management decisions across San Francisco.

SFpark.org/resources/sfmta-on-street-parking-management-policies/

6. SF*park* pricing policy: off-street parking

This document contains the approach to setting rates in SFMTA SF*park* garages, including time of day hourly rates, day of week (i.e., weekday vs. weekend), and other discounted rates, including the methodology for setting those rates. The goal is to communicate a simpler, transparent and data-driven approach for setting parking rates.

SFpark.org/wp-content/uploads/2012/04/SFpark-off-street-pricing-policy-3-29-122.pdf

7. SFMTA parking sensor data storage and communication standards

These are the XML protocols that the SFMTA has developed with parking sensor vendors to receive parking sensor data. These XML standards should enable IT developers to reverse engineer the SF*park* operational data store, reducing the time and money for any city to acquire or develop the technical ability to receive real-time parking data.

SFpark.org/resources/sfmta-parking-sensor-data-storage-and-communication-standards/

8. Complete listing of ODS SQL code (Data definition language)

The following URL contains all the code SFMTA used to deploy its operational data store (ODS). It is over 160 pages and is not included in this document. Because it uses industry-standard SQL language, it should be easily understandable to any DBA, and should be a significant aid in developing a similar system.

SFpark.org/resources/complete-listing-of-ods-sql-code-data-definition-language/

9. XML Reconciliation process

This section provides a bibliography of key detailed project references available online. This document explains the meter XML reconciliation process.

SFpark.org/references/XML_Reconciliation_Process_Documentation

10. SF*park* pilot project evaluation

The SFMTA completed a rigorous evaluation of the SF*park* pilot project. That comprehensive evaluation can be found at:

SFpark.org/resources/docs_pilotevaluation/

11. Parking garage data guide

This document describes how garage occupancy, usage, and payment data are collected and processed for analysis by SF*park*. It also describes how garage rates and information about garage locations and policies are collected and stored.

SFpark.org/resources/sfpark-parking-garage-data-guide/

Glossary

API – Application Programming Interfaces specifies how software components should interact with each other for a human reader. It will outline routines, protocols, data structures and object classes for using that product, language or data feed.

BI – Business Intelligence BI software, in a broad sense, refers to software programs, suites, or packages that allow an organization to create standard operational reports as well as more advanced analytics that cannot be obtained from the individual operational systems.

COTS – Commercial Off The Shelf software An industry-recognized acronym to refer to a software package which is written and supported by a software vendor and then configured to the specifics of the organization.

CSV – Comma Separated Values is a method of storing tabular data in a plain text form. It is a way to transfer field data between programs and languages by a tab or a comma separating each field.

Data Feed Real-time data flow from data sources to users.

DW – Data Warehouse A data warehouse is a database of combined data from different sources, including operational data, which has been processed and stored in a way that makes analysis possible.

DR – Disaster Recovery is the processes and procedures related to reinstating technology infrastructure after a disaster such as a fire, flood or human error that results in system failure.

ESB – Enterprise Service Bus A software product that is used to implement a data architecture model. Typically it allows different software applications to communicate with each other and supports common API protocols.

ETL In computing, extract, transform, and load, refers to a process in database usage and data that:

- Extracts data from outside sources
- Transforms it to fit operational needs, which can include quality levels
- Loads it into the end target (database, more specifically, operational data store, data mart, or data warehouse)

GUI – Graphical User Interface For SF*park* purposes, it refers to the heavy use of interactive maps in addition to charts and graphs for visualization. One feature of the SF*park* design that is somewhat novel is that the entry point GUI for many uses is a GIS map system and/or a BI system. Many users are using BI without having to possess significant knowledge of a BI tool.

HA – High Availability A design approach that ensures a system will be available whenever a user wants to access it. High Availability means there is very little downtime and few outages.

iSCSI Internet Small Computer System Interface is an Internet Protocol (IP)-based storage networking standard for linking data storage facilities.

JMS – Java Message Service is a tool for sending messages between two or more clients using Java Message Oriented Middleware API. A JMS API has a "JJMS Provider" and "JMS Clients" that the "messages" or objects flow between.

JSON – Java Script Object Notation is an open standard format data interchange format. It is easy for both humans and machines to parse and write. It is commonly used to transmit data between a server and a web application. JSON is built on two structures:

1. A collection of name/value pairs. In various languages, this is realized as an object, record, struct, dictionary, hash table, keyed list, or associative array.
2. An ordered list of values. In most languages, this is realized as an array, vector, list, or sequence.

MMS – Meter Management System is an interface system with the various meter vendors to configure any changes to meters wirelessly to the meters. Any event triggered by a meter, such as payments, is sent to the data warehouse.

NAS – Network-attached storage is file-level computer data storage connected to a computer network providing data access to a similar group of users or clients. NAS not only operates as a file server, but is specialized for this task either by its hardware, software, or configuration of those elements.

ODS – Operational Data Store An operational data store is a generic term to describe any source of mostly raw unprocessed data.

OLAP – Online Analytical Processing For purposes of this document, the term is included to distinguish between online transaction processing systems by their need to provide for larger more complex queries and often with fewer users. In more recent years, the terms data warehousing and business intelligence are often used somewhat synonymously.

OLTP – Online Transaction Processing For purposes of this document, the term is included to distinguish the high volume, small transaction size characteristics of typical operational systems from OLAP systems.

Presentation Layer End user applications and/or software tools such as Excel, SPSS, R, ESRI Analytics, Tableau and other reporting tools.

PMR Parking Meter Repair technicians

REST – Representational state transfer is a software architectural style consisting of a coordinated set of architectural constraints applied to components, connectors, and data elements, within a distributed hypermedia system. REST focuses on the roles of the components and their constraints on interacting with the other components. ignores the details of component implementation and protocol syntax in order to focus on the roles of components, the constraints upon their interaction with other components, and their interpretation of significant data elements.

SMS – Sensor Management System is the system which houses and manages sensor data and performance. In the SF*park* example, parking sensors sent data to the StreetSmart sensor management system via a network of pole-mounted repeaters and gateways. The StreetSmart servers then transmit the data to the SF*park* data warehouse.

VMS – Variable Message Sign is an electronic traffic sign displaying timely information to travelers about special events such as traffic incidents, congestion, roadwork or real-time parking availability.

VMS – Vendor Management System is a system to aggregate data and communicate with vendor supplied field equipment, such as for parking sensors or parking meters.

XML – Extensible Mark-Up Language is a markup language that defines a set of rules for encoding documents in a format that is both human readable and machine readable. It is used to clearly mark and organize the different parts of a document so that it can be read on different computer systems. XML is used in both documents and in web services.

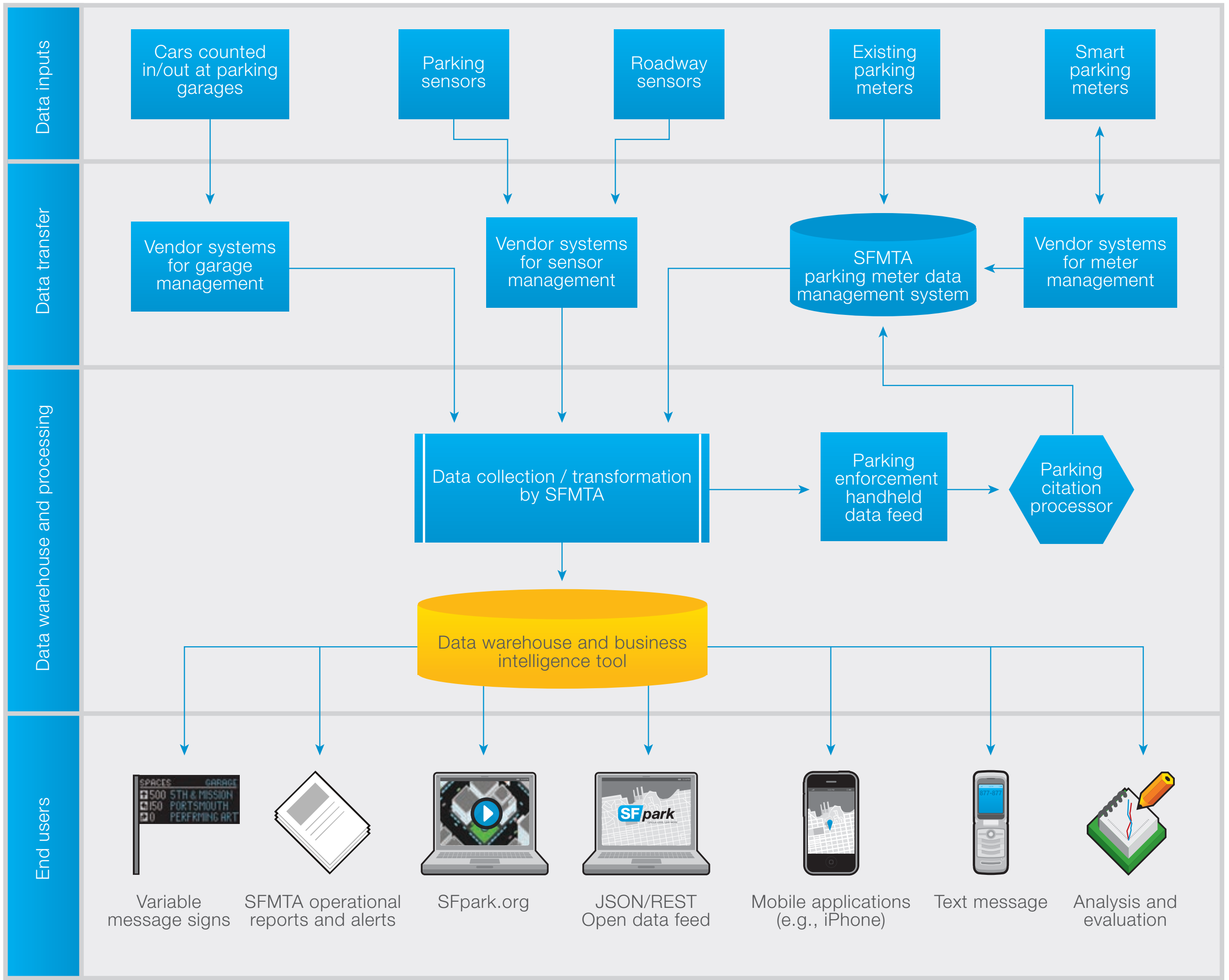
System overview

High level system design

To manage parking effectively, data in an up-to-date and easily usable format is required. SF*park* staff along with Oracle consultants created a parking data warehouse with various data inputs.

The data warehouse enabled SF*park* staff to analyze parking occupancy to make data driven pricing decisions, provide real-time parking availability information to the public, manage the city's on-street spaces, and monitor the performance of the meter, sensor, and garage vendors. It not only stored, but also normalized incoming data, addressing any inconsistencies. The data were used not only for collecting and analyzing parking occupancy rates but also for enforcement management decisions and general parking management agency-wide.

The system diagram below outlines the relationship between the data sources, and end user. There are sensors, meters, and garages that collect and send data to their respective external management systems (operated by the vendor of each product or service), which were then sent to the SFMTA's internal data hub. The SFMTA's SF*park* data management system then made that data ready to be stored in the operational data store, which both pushed out a real-time data feed as well as stored the data for later use by business analysts.



2. PROGRAM MANAGEMENT GUIDANCE

This chapter provides a high level summary of key considerations for program managers and IT project managers when planning and implementing smart parking projects that require complex IT development and integration.

Project planning

Anticipating issues when planning for the IT-related aspects of a smart parking management program

Developing IT-intensive aspects of a smart parking management program will likely present novel challenges for most municipal agencies. This section summarizes key considerations that other cities may find helpful when planning an IT-intensive smart parking management development effort.

Organizational long-term vision

- Undertaking a new approach to parking management can be an opportunity for a city to take a large step forward in terms of how it uses data to make more informed decisions for parking or transportation management. Even if the full ability of the system is not developed during the initial effort, knowing the longer term vision up front is critical when scoping the IT development effort and doing more detailed system architecture planning. This will make it easier to add major enhancements in the future and help to leverage the initial investment.

Clear policy guidance

- When planning for and developing the IT aspects of the parking management system, it is critical to have parking management policies that are clear, unambiguous, and stable. This provides clear direction for the project implementation and technical development teams, and facilitates development because it is easier to translate clear policies into clear business rules which are much easier to IT developers to program.

  The value of clear business rules is subtle but important because they also inform decisions that IT developers make in setting up the initial architecture of their data structures and data models, which would be costly to redo later. For example, the SF*park* technical team's overall approach to organizing and then programming meter configuration data had implications for whether or not the SFMTA could later easily adjust or improve its parking management practices and policies. To provide one specific example, care was taken to organize meter configuration data to later be able to allow customers to pay the meter the night before for the following morning.

Inter-disciplinary teams

- During development planning and development, it is critical to have the business analyst and policy team working closely with the technical development team. For the SF*park* project, successfully specifying the needs of the project and then implementing those technical capabilities required close coordination and deep involvement in each other's work. Rather than having either the policy team or technical team leading the process, the two worked and moved together fluidly. This flexibility and coordination was a strength of the SF*park* technical development effort.

Flexible timelines and resources

- Smart parking management projects with significant data management and complex IT components have a relatively high degree of risk and discovery because they represent new capacities and large steps forward in technical sophistication for most public agencies that manage parking. Program managers should plan for this from the beginning, should not agree to unrealistic project schedules or budgets, and should build in sufficient resources and contingency to allow for necessary research and discovery.

Culture change

- Changes to how parking is managed will likely require some cultural change within an organization. This will require time, resources, attention to change management, and ideally strong relationships with key internal stakeholders. Plan on dedicating time and resources in this area.

Labor and software expense management

- A smart parking management effort akin to SF*park* will require a significant amount of funding and resources. Much of the technical expense of developing the SF*park* data management system came from licensing powerful but proprietary (i.e., Oracle) software tools, investing in in-house hardware, and hiring highly qualified developers to reduce project risk.

  Other cities can likely reduce some of the development effort required by SF*park*. A significant portion of the IT labor expense for SF*park* was in working with parking sensor vendors to resolve technical issues of that new technology (including building monitoring tools for parking sensor data feeds to better manage that contract). Another major expense was managing data in a way that would facilitate the SFMTA's and FHWA's rigorous evaluation of the SF*park* pilot projects, which necessitated a more involved data management and analytical system than other cities might choose to develop. This amount of data processing may not be necessary for other cities that pursue more operationally-oriented systems.

  Another way that other cities can lower costs significantly is by reducing other requirements of the technical system. One example is having fewer vendors plug into the data management system. For example, for SF*park* the SFMTA had to insert data from three meter vendors (in addition to multiple parking sensor vendors, roadway sensors, garage data, and other data sources). Other cities can significantly reduce that expense by having only one meter vendor.

  For other cost areas, it is unlikely that staff costs can be reduced significantly because highly qualified and well-compensated technical staff is essential. A city may be able to reduce hardware-related expenses by using cloud-based solutions. Software costs may be reduced dramatically (while likely increasing development/staff costs) by developing some layers or tools using open source software instead of licensing software for most layers of the data management system, but increases risk and presents other issues discussed later in this document.

Project champions

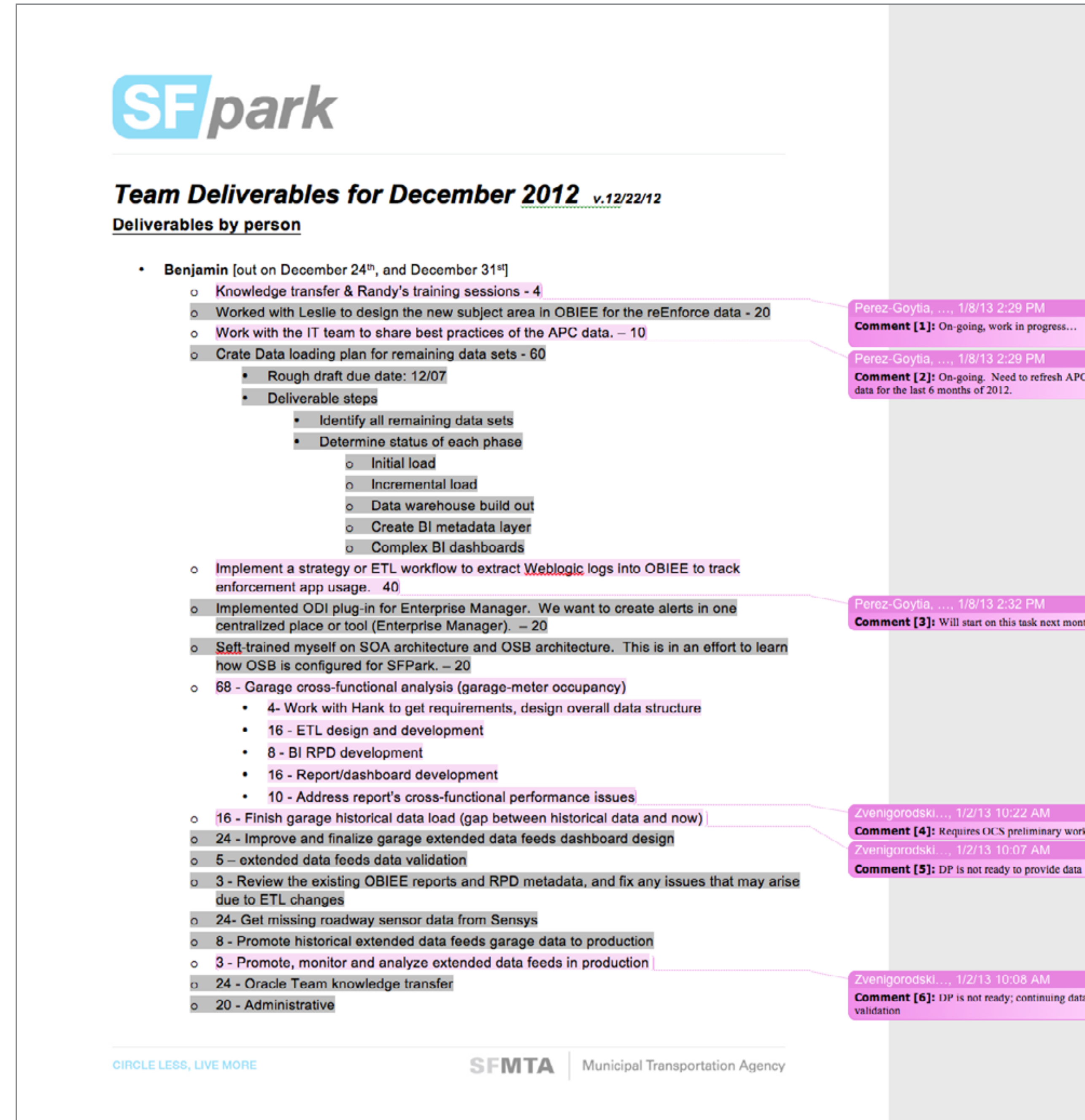
- The schedule for the technical development aspect of a smart parking management project will likely be longer than expected, and will have a higher than typical degree of risk and uncertainty associated with inevitable discovery as an organization learns. For this reason, program managers need to manage expectations of key stakeholders, and those stakeholders and project champions may need to be more involved in this than for other projects.

Methodologically, much within the above bullets are in keeping with the latest best practice approaches to development, including Agile methodology, iterative development, sprints, and extreme programming (whereby business analysts and programmers are paired). We found (and believe) that the more traditional "waterfall" approach (e.g., using Gantt charts) was not and perhaps cannot be successful for a development project with such significant discovery and new technology. The classic waterfall methodology and PMBOK® (Project Management Book of Knowledge) were not abandoned; we still used these methodologies to assist in high-level scope definitions, work breakdown structures, and stakeholder identification, and also to talk about implementation and post-implementation activities.

Below is a table describing which project management tools we used for SF*park* and a sample monthly sprint for IT staff. The sprint is a tool from the Agile development methodology and is a lightweight (ie, simple and not time consuming) way of organizing efforts and tasks. The sprint helps to prioritize and clarify tasks and deliverables by breaking projects into discrete tasks and steps, and helps staff to agree on workloads to feel less overwhelmed. A key strength is that it documents team and individual progress while being flexible within the month as well as month to month and adapt to shifting priorities. Each person writes their own sprint monthly, with input from supervisors and is reviewed at the end of the month to set priorities and to spread tasks and workload appropriately.

Simplified waterfall	Agile development	PMBOK®	
Project initiation		Initiating	Monitoring & controlling
Planning and analysis High level requirements gathering		Planning	
Construction iterations	Design Development Integration and testing	Executing	
Implementation and installation Operations and maintenance Project closure		Closing	

Here is an example of a monthly sprint for technical staff. In this example the grayed items have been completed, items that have not been completed have an explanatory note, and the numbers are the expected hours to complete each task.


SF*park*

Team Deliverables for December 2012 v.12/22/12

Deliverables by person

- **Benjamin** [out on December 24th, and December 31st]
  - Knowledge transfer & Randy's training sessions - 4
  - Worked with Leslie to design the new subject area in OBIEE for the reEnforce data - 20
  - Work with the IT team to share best practices of the APC data. – 10
  - Crate Data loading plan for remaining data sets - 60
    - Rough draft due date: 12/07
    - Deliverable steps
      - Identify all remaining data sets
      - Determine status of each phase
        - Initial load
        - Incremental load
        - Data warehouse build out
        - Create BI metadata layer
        - Complex BI dashboards
  - Implement a strategy or ETL workflow to extract Weblogic logs into OBIEE to track enforcement app usage. 40
  - Implemented ODI plug-in for Enterprise Manager. We want to create alerts in one centralized place or tool (Enterprise Manager). – 20
  - Seft-trained myself on SOA architecture and OSB architecture. This is in an effort to learn how OSB is configured for SFPark. – 20
  - 68 - Garage cross-functional analysis (garage-meter occupancy)
    - 4- Work with Hank to get requirements, design overall data structure
    - 16 - ETL design and development
    - 8 - BI RPD development
    - 16 - Report/dashboard development
    - 10 - Address report's cross-functional performance issues
  - 16 - Finish garage historical data load (gap between historical data and now)
  - 24 - Improve and finalize garage extended data feeds dashboard design
  - 5 – extended data feeds data validation
  - 3 - Review the existing OBIEE reports and RPD metadata, and fix any issues that may arise due to ETL changes
  - 24- Get missing roadway sensor data from Sensys
  - 8 - Promote historical extended data feeds garage data to production
  - 3 - Promote, monitor and analyze extended data feeds in production
  - 24 - Oracle Team knowledge transfer
  - 20 - Administrative

Perez-Goytia, …, 1/8/13 2:29 PM
Comment [1]: On-going, work in progress…

Perez-Goytia, …, 1/8/13 2:29 PM
Comment [2]: On-going. Need to refresh APC data for the last 6 months of 2012.

Perez-Goytia, …, 1/8/13 2:32 PM
Comment [3]: Will start on this task next month.

Zvenigorodski…, 1/2/13 10:22 AM
Comment [4]: Requires OCS preliminary work

Zvenigorodski…, 1/2/13 10:07 AM
Comment [5]: DP is not ready to provide data

Zvenigorodski…, 1/2/13 10:08 AM
Comment [6]: DP is not ready; continuing data validation

CIRCLE LESS, LIVE MORE SFMTA | Municipal Transportation Agency

Roles and responsibilities

The right people for a pioneering project

This section provides the generalized job descriptions for the information system development effort. The SFMTA needed people with a unique blend of skills to form the right team to develop and implement this project.

Program Manager
The program manager oversees all of the projects to ensure the goals of the program will be met. Project managers generally report to the program manager.

Project Manager
The project manager is the person responsible for carrying out the project objectives and ensures all tasks are complete and are in line with the project scope, timeline, and budget. There were multiple project managers in different spheres or disciplines, with project managers for parking meters, parking sensors, and an IT project manager, all working together.

Project Sponsor/Champion
A project sponsor or champion is someone who has a high degree of responsibility and influence in the organization who can advocate for the program and when needed move resources to meet goals and protect the integrity of the project. This could be the director of the organization or bureau, council person, high level manager, deputy director, or someone in a similar position.

Business Owner
A business owner is someone who is responsible for reporting on the success of a specific piece of the parking management program or initiative. This person is responsible for defining program performance metrics, measuring, and reporting those measurements of their area or project.

End User
These are the analysts and technicians who use the data but who are not engaged in project development.

Business Analyst
Business analysts have subject matter expertise in a specific agency or organizational business area, such as parking meters or parking garage data systems. They use technology to assess and improve business operations.

Database Administrator
A database administrator is the person responsible for developing, installing, configuring, and maintaining the databases for a program. They will also make sure that data formats are compatible and functional for the project.

System Administrator
The system administrator is the person who maintains and configures the reliable operation of computer systems. They manage software and hardware for the system and provide support and troubleshoot for project managers.

Change Management Coordinator
This person is the gatekeeper for any vendor system version changes or internal system changes. They assess and minimize risks by controlling the timing and scope of changes to all systems (if only system upgrades and maintenance).

Parking Inventory Manager
Someone needs to keep track of and maintain the parking inventory in the data warehouse. In the SFMTA, this person coordinates with the Meter Shop and enters any changes to the system. The responsibility for keeping meter data up to date was gradually transitioned to the Meter Shop, with the parking inventory manager assuming a quality control role.

External Vendor Points Of Contact
For each vendor, it is important to have a single point of contact for project managers to interface with and to ensure quality vendor performance.

Implementation approach

To buy or to build

For the SF*park* pilot project, the SFMTA had an ambitious vision for a relatively complex parking management approach and the system that makes it possible. Most broadly, for IT-intensive parking management projects, organizations need to decide at the outset what approach to implementation to take and how much of the development (and learning) to take on in-house.

SF*park* contracting approach

The SFMTA had several broad options for contracting when considering how to accomplish the SF*park* pilot project. At one end of the spectrum, the SFMTA could have attempted to develop and manage the program entirely in-house, developing everything for the project with its own staff, such as using open source tools to build all of its own data acquisition and business intelligence tools, and implement the program entirely without contracted services or goods. At the other end of the spectrum (as was chosen in Los Angeles for their Express Park program), the SFMTA could have chosen to deliver the entire program through a turnkey approach, hiring a prime contractor to develop and manage the entire program.

The SFMTA used elements of both approaches, contracting some services but maintaining control of day-to-day management and design in a way that fit the SFMTA's needs and capacity. In this hybrid model, the SFMTA bought the smaller pieces of software (such as the business intelligence or analytical software that allows for reporting and ad hoc analysis) but chose to build the data integration piece or layer of the system. This platform allowed the SFMTA a high degree of control and made it possible for the SFMTA to then easily add or test different vendors (meters, sensors, pay by phone, enforcement, transit, etc.) so long as they could meet basic protocols for data structures and communication. That portion of the system is also what allows the SFMTA to do more effective contract management, operation, evaluation, report writing, ad hoc analysis, etc.

Reasons to pursue this hybrid approach include:

- The SFMTA chose to develop a data management platform of its own so that the SFMTA, rather than a vendor, would own and manage that platform. By creating this agnostic system, the SFMTA can now simply add or remove components such as parking meters, parking sensors, or other data sources. This prevented dependence on any vendor.
- Developing this system was a major organizational challenge, but it also enhanced the SFMTA's knowledge and ability to manage parking and the overall transportation system in the future.
- Understanding and interpreting parking and transportation data is a core part of the SFMTA's business. There was no off-the-shelf tool available to integrate multiple parking-related feeds (e.g., meter, sensor, pay by phone, and garage) from multiple vendors, so the SFMTA chose to develop those tools in-house by modifying existing software.
- Developing a somewhat custom data management system allowed the SFMTA to meet the evaluation needs of a federally-funded demonstration project.
- The SFMTA chose to purchase software tools rather than develop its own. While open source software tools were considered, extensive software development is not one of the SFMTA's core competencies, especially for data acquisition tools.

Whether buying or building, it is critical that all choices are consistent with the current technical standards as well as future strategic direction of your organization's IT group.

Building organizational capacity

Developing some portions of the pilot project technology in-house was an opportunity to build SFMTA staff capacity and organizational expertise:

- Being heavily involved with the hands-on, day-to-day contract specification, procurement, and subcontract management process was a way to deepen the SFMTA's familiarity with various aspects of the SF*park* pilot technology.
- Playing a strong role in contract and subcontract management was another opportunity to increase SFMTA staff capacity in a way that would have been minimal in a turnkey solution.
- The SFMTA contracted out some key portions of the project. For example, the SFMTA wanted to employ known and proven experts for sophisticated IT development to plan and develop a data warehouse and business intelligence tool—the SFMTA did not have that skill set in-house.
- Some of the skills required to build the SF*park* data management are very specialized, and are not typically found in IT staff at most municipalities, so it is likely that agencies will need to hire some contractors and consultants. The SF*park* development process was an opportunity to increase the SFMTA's overall information technology sophistication and begin to apply those lessons to other parts of our agency and operations.
- To provide the necessary staffing levels for the large and temporary level of effort required to plan and implement the pilot project, the SFMTA augmented its internal staff with contractors who were part of the project team.
- Because some IT skills likely need to be contracted, and as the exact technical and skills needs may be discovered or clarified as part of the technical development process, an IT project development team will likely need contracting support. For SF*park* it was useful to have a contracting vehicle available that enabled the team to rapidly get necessary specialized outside technical assistance.
- When planning for this IT development, the staff and skills necessary for the designing and building of the system are somewhat different than those required to maintain and enhance the system. In-house IT staff can likely (possibly with some training) be expected to maintain the system, but specialized (but temporary) outside support is almost certainly needed for its development.

Operational considerations

Operational challenges must be confronted

Although the objectives of the pilot project were focused on the analysis, operational needs could not be ignored. Since a working inventory management system was built to manage the sensors and smart meters, the operational data flowing in required programming to allow for the operational management of this data.

Considerable time and expense were invested in creating systems to manage the operations and inventory of these assets and allow for multiple vendors of each asset. Since multiple vendors were used, it was obvious that each vendor management system would be inadequate and that a consolidated operational data store was needed. Essential to this were the data-feed specifications developed to standardize data flowing into the data warehouse.

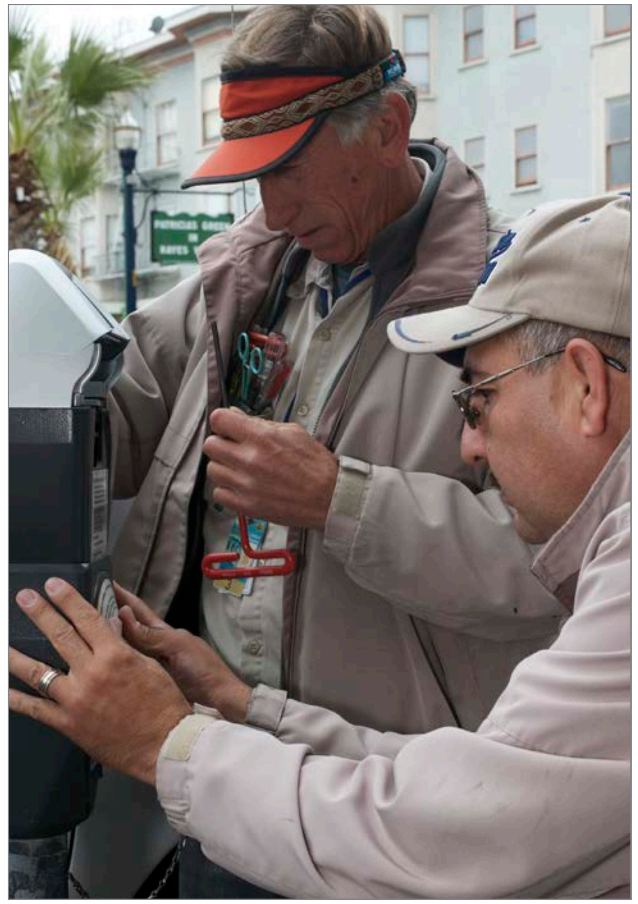
System operations are often-overlooked or underestimated. The questions below are intended to help begin consideration of how to support what has been built once it is launched.

- Asset management: How will changes to the physical devices in the field be handled and understood by all parties?
  - Historically changes to parking meter configuration were an operational process handled by the parking meter repair technicians (PMRs). The introduction of the data warehouse made the existing processes superfluous and also increased the chances that the meter configurations in the vendor meter management systems become out of sync. To avoid inconsistencies between the various databases, the meter configuration process was reassigned to a dedicated staff member who made changes directly to the SF*park* data warehouse rather than having PMRs communicate changes back to the SF*park* staff to make changes to the database. By adjusting the process such that changes had to be communicated through the SF*park* data warehouse it made the likelihood of discrepancies much smaller.
  - Performance alerting and problem detection—How will the SFMTA know when problems happen? When data feeds slow down, often the operators in the field will know first. How can the system help detect and track these issues? How will the SFMTA know if a meter or sensor has stopped reporting? The vendor may have its proprietary system indicate its status, but often that reflects what its central server believes is the status and not necessarily what is its real status on the street. Using the parking meter and sensor data stored in the data warehouse average activity levels were calculated by hour and day type. Reports alerting staff to deviations from these average activity levels were created to warn of any potential equipment problems. False alarms were often due to holidays, special events, or any other events which could affect average parking patterns.

- Contract administration: How will the city measure performance and develop metrics so that contract service levels are met?
  - Performance metrics were established in the contracts of vendors providing data to the SF*park* data warehouse. The ability of the vendors to meet these metrics directly impacted the amount paid by the SFMTA to the vendors.
  - The metrics generally related to the accuracy, timeliness, and latency of the data.

- Change control: How can the city and the vendor(s) communicate about hardware and/or software changes on both sides?
  - Any changes to the systems that interact with the data feeds, whether hardware or software, required approval from SF*park* staff and Oracle consultants to prevent unexpected outages.
  - Timelines, specifications, performance metrics, and emergency contacts were agreed upon by both parties and run through test environments prior to implementation of changes to protect against data loss or service interruptions.

- Meter configuration via XML reconciliation: Do the meter vendors processes in place to accept and apply updates from the SF*park* data warehouse automatically? And are the configuration details compatible?
  - There are many ways to structure the same information. Meter vendors have generally operated in a space where they store data in a way that is suitable for their needs. SF*park*'s time of day and demand responsive rate adjustments challenged vendors, and as a result forced them to adapt to dealing with more complex schedules. Meter vendors found ways to translate configuration detail between the two databases, though in some cases detail was lost.
  - A reconciliation process was built to analyze any reconciling items between the two databases, determine whether or not those differences would impact meter operation, and make corrections if needed. Please refer to the reconciliation process document listed in the references for details.



Contracts

Procuring necessary goods and services

Implementing the SF*park* pilot project required contracts with a variety of vendors as well as working with these vendors to confront project challenges together. The contracts for those goods and services, as well as the management and oversight of those contracts, were critical for delivering this technical effort.

Contract management considerations

During procurement

- Clearly specify the deliverables and performance standards (ideally with financial incentives to achieve targeted performance), including the methodology for measuring them, in RFPs and contracts.
- Do not limit yourself with unnecessarily specific statements of work. Focus on the business deliverable rather than a specific technology implementation, as the technology will change.
- Relate payment to performance for both upfront and ongoing deliverables.
- Relate payment to delivery of functionality and not delivery of equipment.
- Rigorously field test and verify all vendor claims and performance in real-world tests during the selection process.
- Have prospective vendors submit sample project plans, support procedures, and service level agreements with their proposals.
- Avoid getting between a vendor and another agency. Contracts should be structured so that the vendor is directly responsible for any permitting required by other agencies.
- Anticipate that project contracts for unfamiliar and complex technology and services will require a large amount of legal support.

During contract management and delivery

- Include contingency that is adequate and appropriate for a new undertaking in a complex area.
- Always have “plan B and C” strategies.
- Rigorously field test and verify all vendor claims and performance in real-world tests during the formal acceptance of the goods or services.
- Require that all backup and redundancy/high-availability features be demonstrated as part of the acceptance process.
- Put in place data management and reporting to make it easy to manage performance-based contracts.

3. SYSTEM ARCHITECTURE

This section provides a high-level conceptual model of the SF*park* system that describes both the structure of the system's hardware and software components as well as the behavior of the system software. As this information is quite technical and detailed, the intent of this section is not to be exhaustive, but to provide a framework for understanding the underpinnings of the SF*park* system.

System hardware architecture

The physical technical foundation of SF*park*

Creating the SF*park* data management system and then preparing to run a real-time information service challenged the SFMTA to determine the best ways to create, support, and maintain a system with the rigor that is required for providing a high-availability data service. What follows is a description of the choices SFMTA made.

The SF*park* system included many hardware system components internally and externally including:

- Servers
- Communications networks
- Databases
- Data warehouses
- Application Programming Interfaces (APIs)
- Parking sensors
- Smart parking meters
- Garage entry and exit counters
- Smart phone apps

The SF*park* system was implemented using HP servers and an EMC storage area network (SAN). The choice of these products and vendors was based on existing SFMTA standards, as well as the very significant challenge of building such a comprehensive data warehouse.

To support the evaluation portion of SF*park*, the SFMTA built several data marts. Other agencies may have a more operational focus and not have elaborate requirements for evaluation or reporting as with SF*park*. This would significantly simplify the size and complexity of the architectures.

During the phase of hardware architecture design and vendor selection, a city should consider both high availability (i.e., the percentage of the time that the system is available and operational) as well as disaster recovery needs. Will the system truly be needed 24 X 7? Is the system considered mission critical? What is the acceptable recovery time for various functionalities of the system? In short, how long can the system be down?

These detailed SF*park* system diagrams illustrate the architecture and data flows.

- SF*park* data flow, please see page 43 for diagram.
- SF*park* handheld architecture, please see oversize insert
- SF*park* physical architecture, please see oversize insert

Vendor hardware architecture

Sensors

Overview

For parking occupancy information from metered spaces, SF*park* relied on wireless in-ground parking sensors that detect vehicle arrival and departure. StreetSmart Technologies provided parking sensors for SF*park*. One or two sensors were installed in each parking space. Each sensor has a magnetometer that looks for changes in the earth's electromagnetic field and is calibrated to detect vehicles in the surrounding area. StreetSmart's sensor is an in-ground battery powered wireless device. The sensors sent data to the StreetSmart sensor management system via a network of pole-mounted repeaters and gateways that formed an ultra low power mesh network and communicated, via GPRS modem, data to StreetSmart servers to in turn process the data. The StreetSmart servers then transmitted the data to the SF*park* data warehouse. For more detailed information on sensors please refer to the parking sensors performance standards document listed in the references.

Technological opportunities and limitations

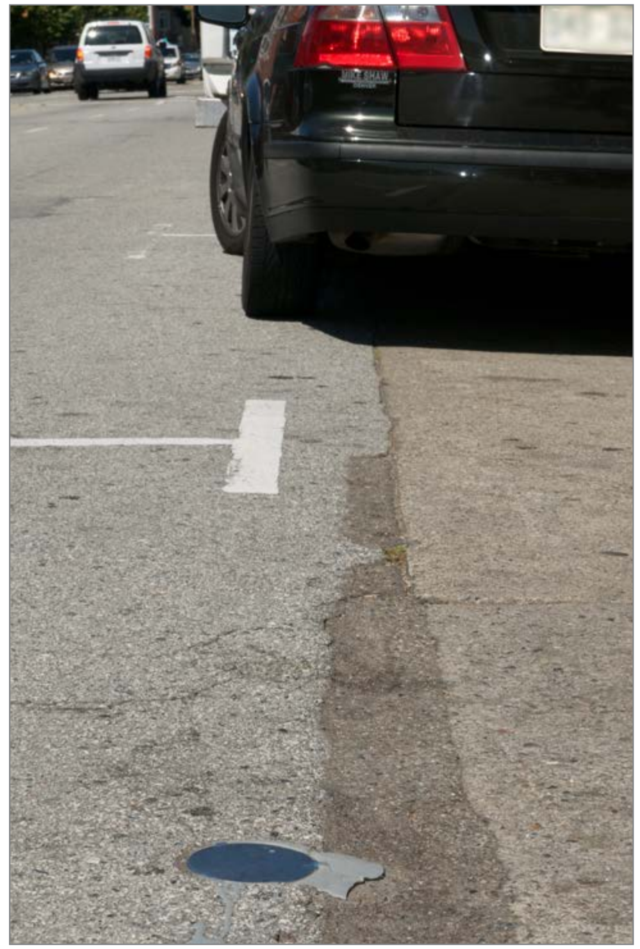
- In an urban environment, many things emit an electromagnetic field, including utility boxes and the overhead lines that power electric buses. Such sources of interference can reduce the ability of sensors to accurately detect arrivals and departures.
- Vendors are developing filters for this interference, but every block and neighborhood is different, and can pose new challenges for the sensors. In areas with high levels of interference, our vendor installed two sensors per space.
- Verifying sensor data is a labor-intensive process involving manual verification of sensor events and accuracy. Automated data collection tools can improve the efficiency of field surveys. Sometimes there are tradeoffs between the latency and accuracy of parking sensor data.
- Sensors are fine tuned to sense vehicles in a defined area, and clearly demarcated parking spaces have been necessary. Sensors in undemarcated areas (e.g., residential parking areas or pay-and-display applications) are not yet useful in our experience. Sensors are also less accurate at detecting small cars or vehicles with fewer than four wheels.
- Sensors and their associated equipment are new devices out on the street. Other agencies, utilities companies, and vendors that regularly perform street work may inadvertently damage or destroy them. It is important to coordinate with other organizations and educate them about the sensors. Also, define the process for addressing these issues in the contract.
- Keeping an accurate geographic inventory of where all sensor-relate devices are located, down to the latitude and longitude coordinate level, makes them easier to manage.
- Solar-powered equipment is much easier to set up and operate than equipment that connects to the electrical grid.

Parking sensor performance standards and measurements

This table outlines SF*park*'s sensor performance standards and the measurements used to test performance.

Sensor performance	Measure	Highest-level standard
Reporting	Sensors must send a message every day, even if the space's status has not changed.	98 percent must report every day in each area. Exceptions are made for spaces where sensors were removed or damaged.
Accuracy	A field observer walks down the street and notes whether spaces are vacant or occupied.	92 percent accurate within each pilot or control area.
Timeliness	This field test involves recording the exact time that a car enters or leaves a space. Receipt of a correct event from the data feed is confirmed, and latency is calculated by how long it took the sensor network to process and send that event	85 percent of events must be received within 60 seconds.

- The timeliness test is the most telling measure of sensor performance. With the data accuracy test, the sensor vendor has a 50 percent chance of being correct (because a space is either occupied or vacant, simply reporting one of the two possible states has a high probability of being correct). The timeliness test involves verifying each parking event we receive.
- The timeliness test is time consuming because the surveyor needs to wait for parking events, each at different spaces. Data collection rates are variable, but average about 10 observations per hour; about 100 observations per hour are collected with the data accuracy test.



A parking occupancy sensor installed in the street

Meters

Overview

The parking meters used in SF*park* accept credit card payments, transmit payment data (for all payment methods) to a central server in real-time, and allow new rates and display information to be programmed and deployed remotely. Each smart meter is equipped with a cellular radio and SIM-card to communicate with the data warehouse as well as the vendor's meter management system. The data warehouse stores all events triggered by the parking meter, such as payments, and interfaces with the meter management systems of the vendors to push configuration changes to the meters wirelessly.

In addition to the live data feed, SFMTA staff can reprogram meters remotely through the SF*park* data warehouse, which communicates with the vendors' meter management system via XML. The SFMTA has established a meter XML protocol to ensure that regardless of the vendor or model of smart meter all meter communications will be compatible with the data warehouse. Refer to the XML protocol in the SF*park* meter vendor rate change protocol described in the project references for further detail.

Technological opportunities and limitations

There are a host of stakeholders who interface with the meters and the meter management systems in the back end (e.g., public, maintenance personnel, parking control or enforcement officers, finance, customer service, adjudication, and coin collection). Current systems do not necessarily serve all users equally well, and changes to the systems to improve usability to one group may affect usability for another group. It is necessary to rigorously bench and field test new features and adjustments before implementation. Before you have adequately demonstrated the performance of new features and system adjustments, field tests should be kept to geographically small areas so that errors can be easily corrected.

New meter technology relies on cellular communications. If network coverage is inadequate, operations, reporting, and maintenance capabilities are compromised. Most smart meter technology relies on solar power to extend battery life. This presents a challenge for using these meters in parking garages and otherwise shady locations. We used a few different cell providers in order to get the best reception possible in different areas of the city.

A 'smart' SF*park* parking meter

Garages

Garage occupancy data is collected in a few ways.

1. Loop counters

The garage revenue-control vendor keeps track of how many cars are in the garage at any given time using equipment on each entry/exit lane. Entering/exiting cars go through a three-step process:

a. Roll over the "loop counter" (essentially a metal detector) that engages the ticket machine.
b. Hourly parkers press the button for a ticket upon entry, or insert their ticket upon exit. Monthly parkers scan their monthly parker card.
c. The gate goes up to let the car enter or exit the garage, and then down once the car has cleared the gate.

2. Manual car counts

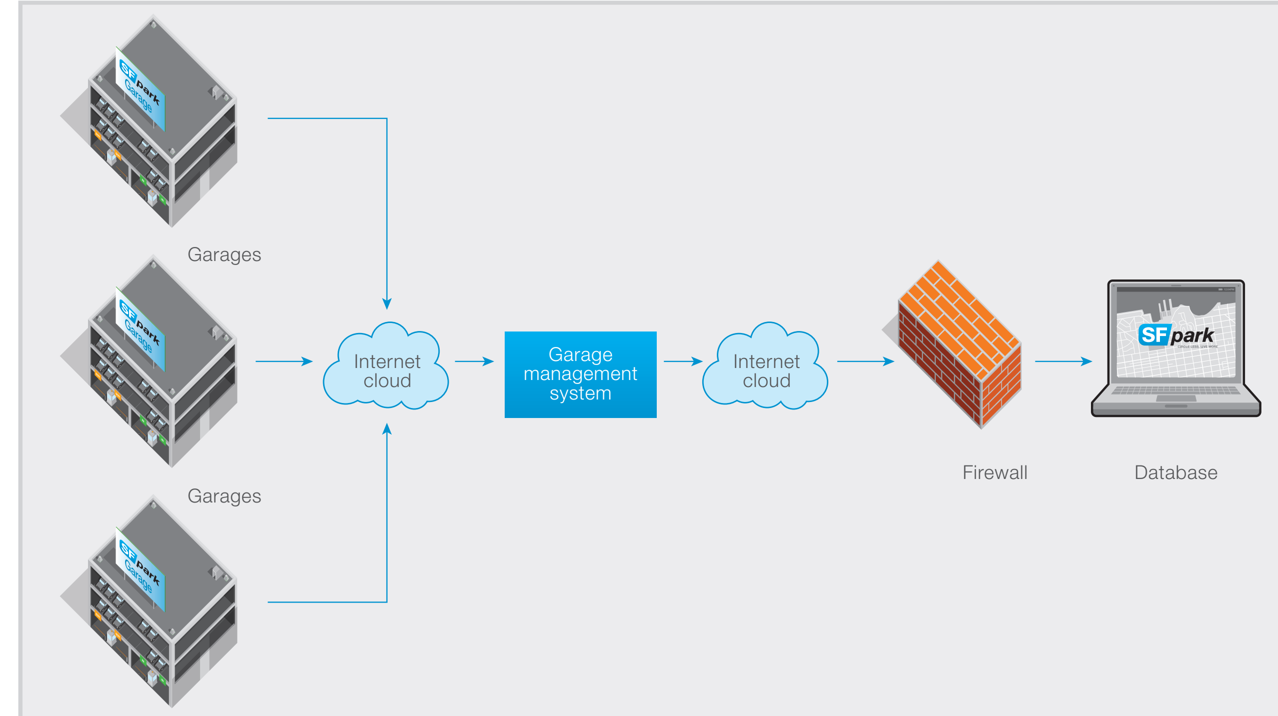
To address the potential introduction of error into the car counts, the SFMTA requires garages to count manually the number of cars in the garage at least once per day, and update the car count to match the manual count. Each manual count update is recorded in the garage system and sent to the SF*park* data warehouse.

3. Garage payment data

Garage usage/payment data is divided into four distinct data feeds:

a. "Session"—entries and exits
b. "Transient payment"—payments by hourly parkers
c. "Monthly payment"—payments by monthly parkers
d. "Statistic"—aggregated data regarding tickets issued, gate entries/exits, etc., used for auditing purposes

The entry and data collection system at an SF*park* garage



Garage data flow

Data feeds

All the above data sources create a data feed which allows the public to find parking availability and price information via mobile and web applications. Web services were the primary method used to collect, store, and transmit real-time data from the sensor providers to the SF*park* data warehouse.

The SF*park* real-time parking availability data feed is publicly available at no cost to application developers and others interested in the data. This data feed is provided as a REST service with data returned in XML or JSON format. SF*park* provides a document containing the information developers need to carry out testing for retrieving and displaying the parking availability data from the SF*park* data warehouse. Refer to the SF*park* availablity service API reference described in the project references for further detail.

System software architecture

Data sources, formats, feeds, and flows

Creating the data component of the SF*park* system and then preparing to run a real-time information service challenged the SFMTA to determine the best ways to create, support, and maintain a system with the rigor that is required for providing a high-availability data service. What follows is a description of the approach that the SF*park* team chose to represent software layers to achieve the data acquisition, processing of data feeds, and data flowing to external sources.

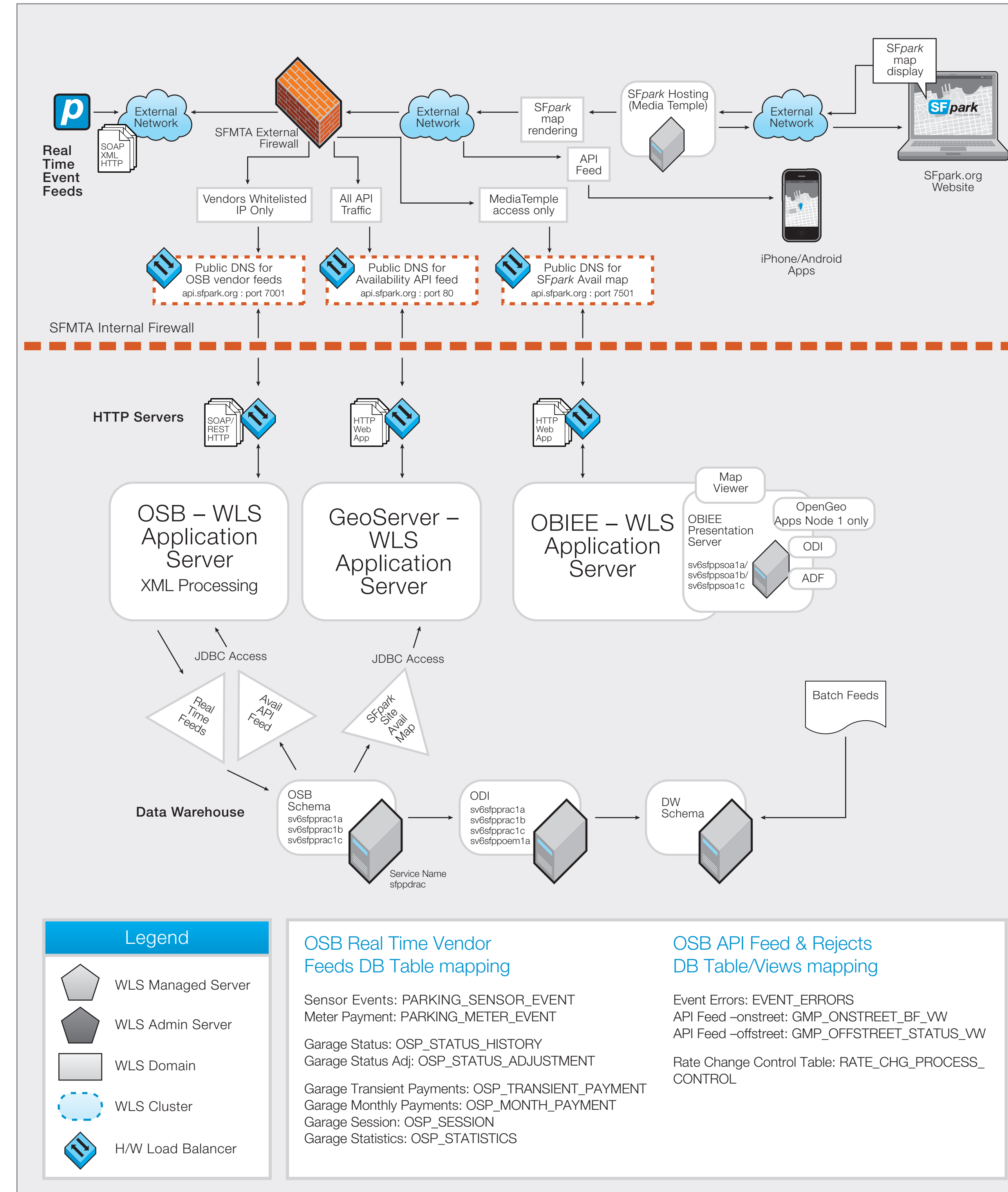
At a high level, the architecture of the SF*park* system, including external interfaces, consists of the following application layers. Acronyms listed below are defined in the glossary. The SF*park* software layers listed below are detailed in following sections.

Vendor

- Equipment (meters, sensors, etc.)
- Data feeds from vendor

SF*park*

- Operational Data Store (ODS)
  - Data feeds processing software through the ESB. Refer to the Complete listing of ODS SQL code document in appendix X for details on the ODS schema and major tables.
  - Data validation
    - Passes validation—insert valid data into ODS
    - Fails validation—return error message and insert invalid data into error logging tables
- Data warehouse
  - Process ODS data with extract, transform, and load processes (ETL)
  - Update metadata about data warehouse structures (if necessary)
- Presentation layer/BI
  - Create and distribute canned reports
  - Create any materialized views and/or aggregate tables
  - Make data available to all supported presentation layers
- Data feeds
  - Availability feed
  - Data push to 511.org
  - Prototype parking enforcement application API for two-way data transfer and validation



SF*park* data warehouse design

Turning data into information and operational knowledge

The SF*park* pilot project required a robust and powerful database, both for operations and to complete a thorough evaluation. This chapter describes the choices the SFMTA made to build the operational data store and data warehouse for SF*park*.

With respect to the SF*park* system, database design involved two quite different disciplines: a transaction-oriented data store and a very differently structured analytical processing data store. The first is designed to be "lean and mean" in order to support high volumes of database read and write operations but relatively low volumes of data. Analytical processing is very different. It typically involves much lower volumes of users but higher volumes of data.

Operational data store

The operational data store, or ODS, is a generic term to describe any source of mostly raw unprocessed data. In this document, it refers to data that comes from external vendor systems such as a meter management system, or from other operational or transaction-based software that already exists within the SFMTA. This data is almost always transactional in nature, with small records and a potentially high number of records. This data is often very challenging to report on beyond simple queries such as "tell me what's happening now" or "show me last month's totals." Many times the reporting and BI needs of the organization will require some data warehousing to relieve the ODS from the processing burden of BI and/or OLAP and allow the system to be more available for OLTP.

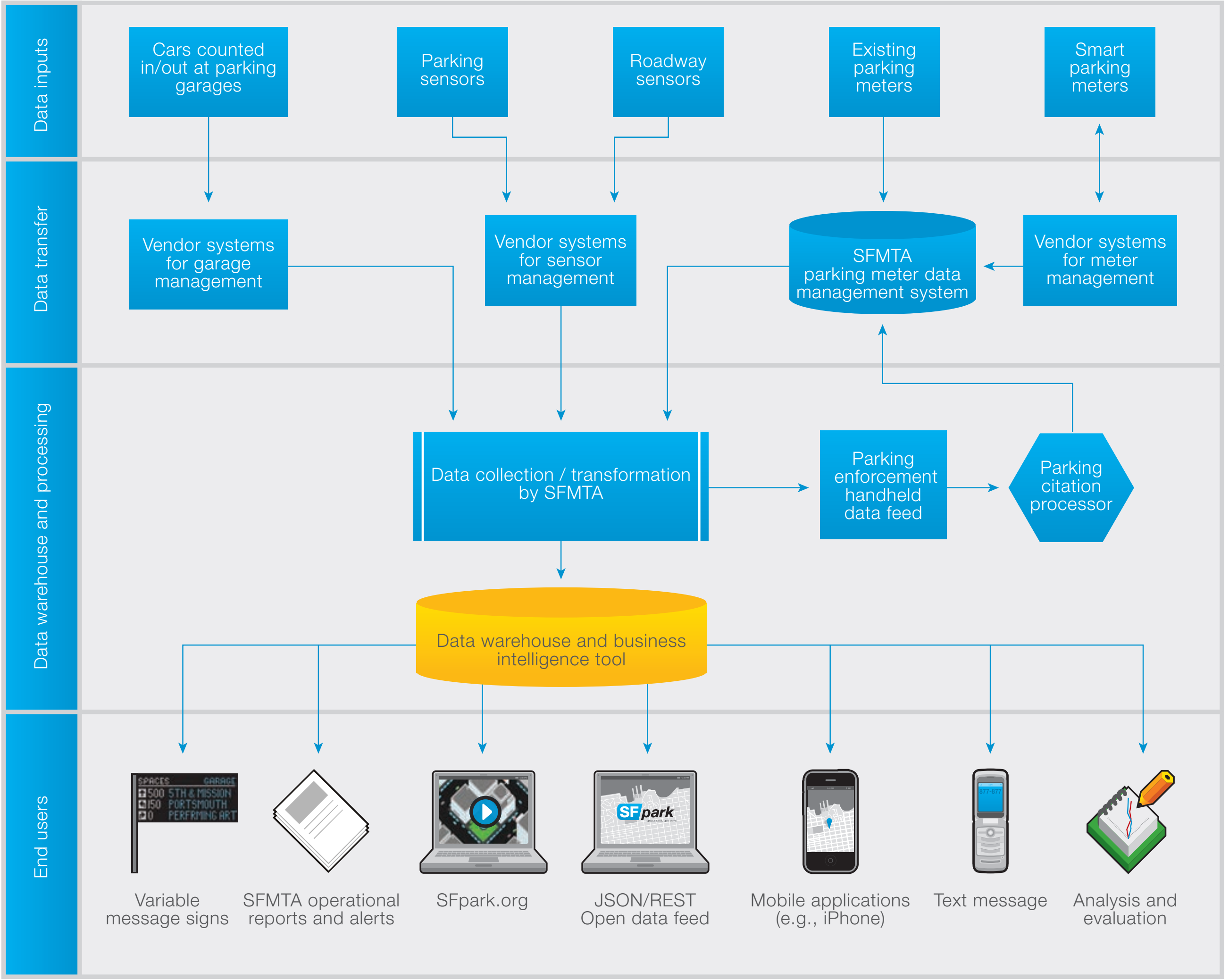
Data warehouse

Data warehousing is a specific discipline. Subject matter experience and expertise in this area are both vital. Individuals with this expertise might have titles such as a Database Administrator or Technical Architect. Other terms like BI developer and Data Warehouse Architect can apply. The distinguishing area of expertise is the ability to analyze operational data and organizational needs, and create dimension and fact tables that provide the data structures to meet those needs with advanced reporting and analytics. Without this restructuring of data, usually called ETL for Extract, Transform and Load, any complex report or analysis might take hours or even days to complete, or even fail to complete.

The SF*park* data warehouse is modeled on the Ralph Kimball approach, and this approach is recommended. Ralph Kimball is widely regarded as one of the original architects of data warehousing and is known for long-term convictions that data warehouses must be designed to be understandable and fast. His approach to data warehousing is known by being bottom up as opposed to top-down, focused on business process oriented small data marts as opposed to one centralized warehouse, and built on dimensional modelling. Given the high degree of uncertainty and newness to technology, using the competing strategy of Bill Inmon, which is characterized by it's top-down design methodology which generates dimensional views of data loaded from a centralized repository, or enterprise data model, is not recommended. Of these two predominant approaches to data warehousing design and development, the SFMTA chose the Kimball approach.

All the SQL code used to create the ODS is provided in the Complete listing of ODS SQL code (Data definition language) document; a link can be found in the project references section on page 14.

This diagram describes how data inputs flow to the data warehouse, and then finally to end users.



Data sharing architecture

How data flows from source to user

At the core of the SF*park* system architecture is a logical data flow from vendors to the operational data store, then to the data warehouse, and ultimately to BI repositories, external agencies, and smart platform apps.

Data feeds are a vital part of the architecture. Typically data flows one way (ignoring error messages) from vendors through the ODS and data warehouse to layers such as BI, other presentation software packages or APIs that deliver data to other systems or applications such as a smartphone app. The SF*park* system used XML as its standard communication protocol for vendor-to-agency or agency-to-agency communication and used JSON where applicable. One application of JSON was to communicate availability data to smartphone applications. Currently we do not deploy JSON for two-way data communications, preferring instead the stronger structural enforcement that XML provides.

It should be noted that records are stored in a persistent ODS schema as soon as possible; therefore these should be transmitted in their proper sequence. This helps maintain data integrity in the SF*park* data warehouse: for example, the session start event record for a particular sensor has been processed before the session end event record. Also, it is recommended for the provider to implement suitable response status and error checking and recovery as well as capability to pause and resume transmissions during any planned or unscheduled maintenance of the SF*park* services.

Parking sensor data feed

- The parking sensor data feed is implemented as a web service that enables the SF*park* parking sensor providers to transmit real-time parking event records to the SF*park* data warehouse. The SF*park* web service implementation utilizes Oracle Service Bus (OSB) which runs on the Java-based Oracle WebLogic Application Server (WLS) for the processing of these records. OSB allows multiple clients to invoke the same service and clustered server instances to be running thereby allowing for high throughput, fault tolerance, and performance.

  Testing should also help verify the response status for various common kinds of errors that may be expected during such transmissions, such as transmissions of duplicate records, network latency or errors, incomplete or invalid format of XML elements, etc.

Meter payment service data

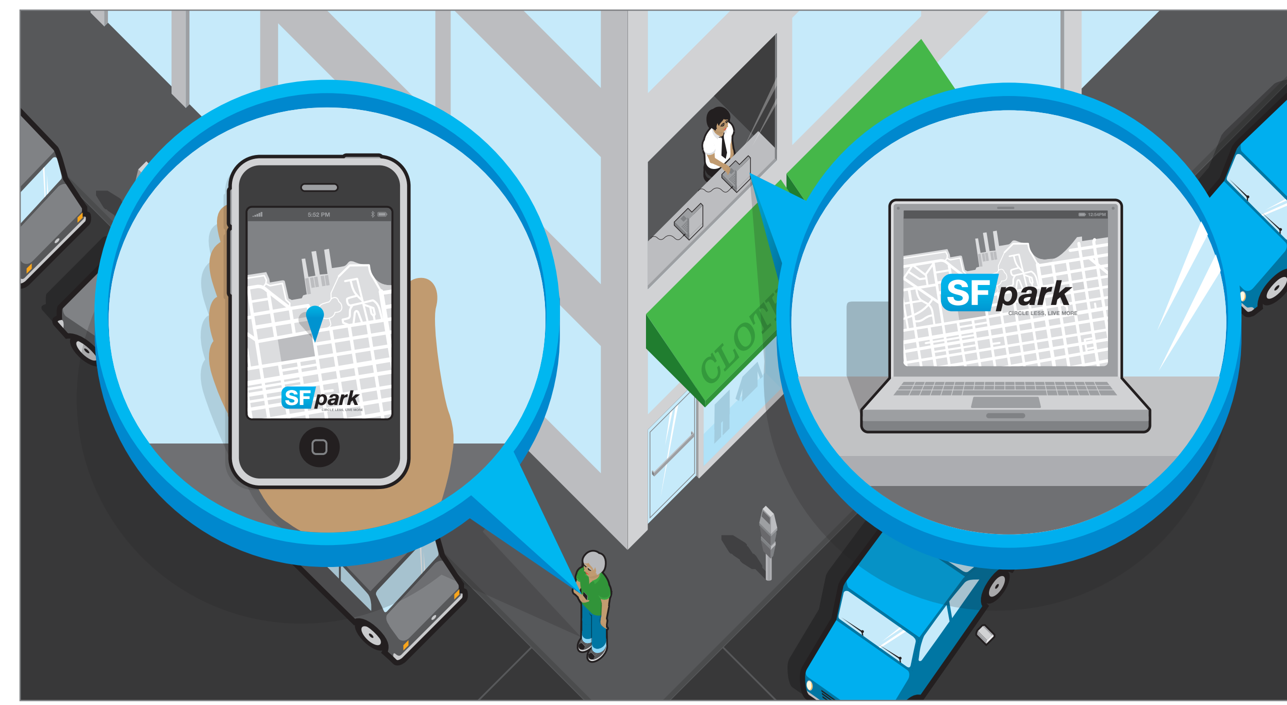
- The meter payment web service enables the SF*park* meter payment providers to transmit a real-time payment event record in XML format for persistence to the SF*park* Data Warehouse. The SF*park* web service implementation utilizes OSB which runs on the Java-based Oracle WLS for the processing of these records.

Garage data web services

- This was the first web service deployed and is also the primary service used to collect the basic garage count information used to display garage availability. The garage web service enables the SF*park* garage providers to transmit a real-time garage count status record in XML format for persistence to the SF*park* Data Warehouse. The SF*park* web service implementation utilizes OSB which runs on the Java-based Oracle WLS for the processing of these records. OSB allows multiple clients to invoke the same service and clustered server instances to be running thereby allowing for high throughput, fault tolerance, and performance.

Smart platform web services

- The SF*park* real-time parking availability data feed is publicly available at no cost to application developers and others interested in the data. This data feed is provided as a REST service with data returned in XML or JSON format. Please see Appendix 3 for the SF*park* availability service API for information for developers on carrying out, testing, retrieving and displaying the parking availability data from the SF*park* data warehouse.



4. USER INTERACTION

One significant challenge of the SF*park* project was to create a system to manage thousands of smart meters and sensors with an easy-to-use, intuitive user interface. It was decided early on to use maps, such as Google Maps or ESRI layers, as the display and portal for entry into the management of data.

Map-based user and programmatic interfaces

Improving the user experience of these tools

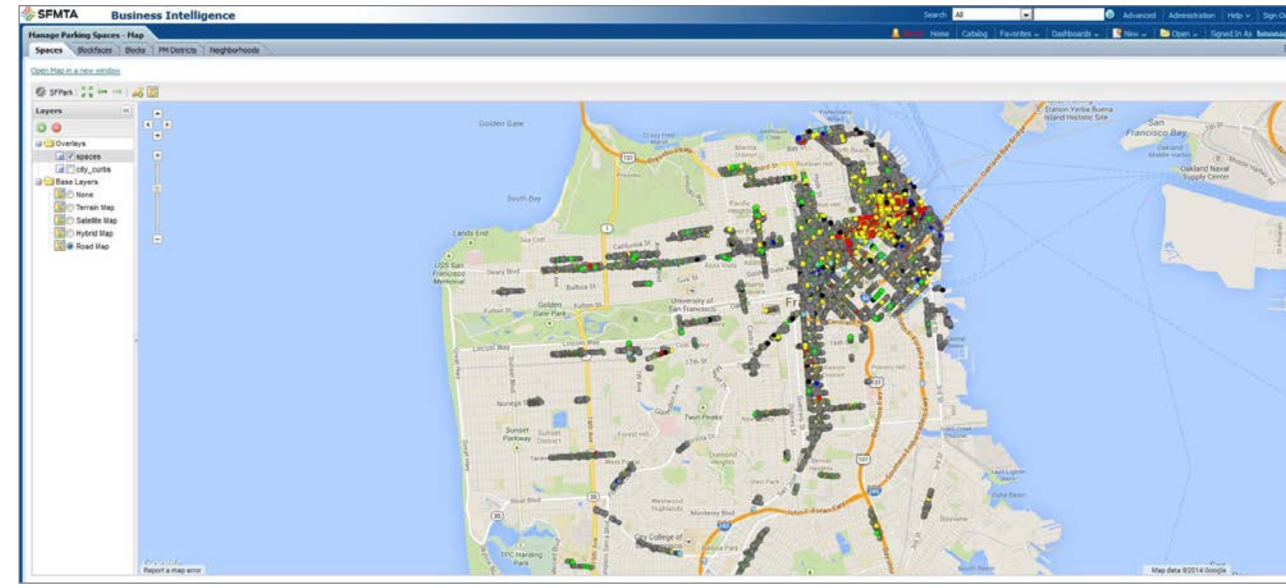
Smart platforms are changing the way applications are being developed. Users now expect touch response and two-way interaction for all applications.

SF*park*'s Parking Space Inventory module was deployed using maps as the entry portal. The first screen a user encounters is an interactive map which allows the technician or inventory manager to "drill down" on a specific location using the zoom function built into most mapping software. This capability proved to be very intuitive and was readily accepted. It was a challenge to develop as the capability to connect the maps to modules that ensure database integrity has a medium-high level of complexity. We believe that this approach was worth the effort and expense.

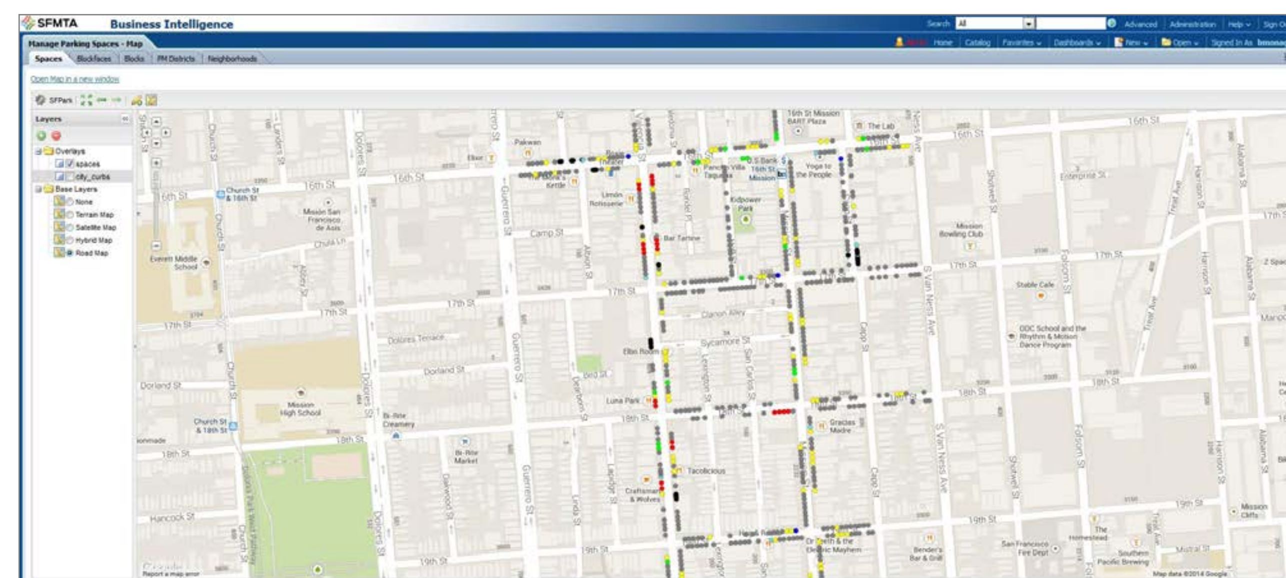
Perhaps this software will become part of future meter and sensor vendor offerings. The trends in all application software seem to be converging on touch screen interaction, access to interactive maps when needed, and two-way interactions with business intelligence software.

The following screenshots are examples of the map-based interface to manage the parking space inventory in BI.

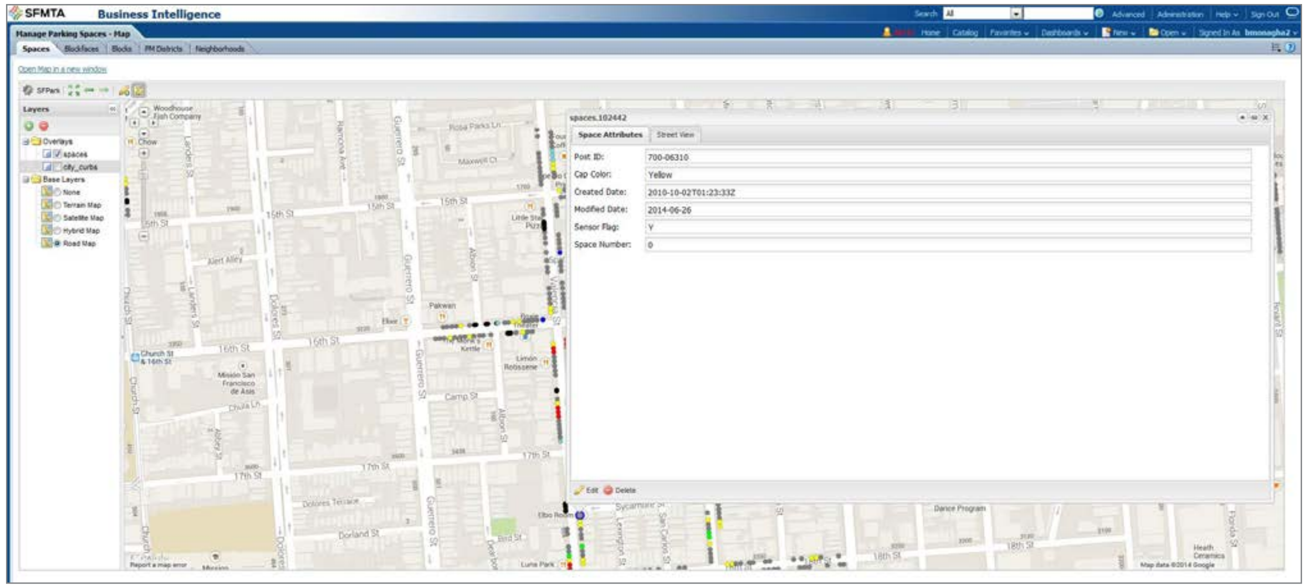
Parking space inventory map



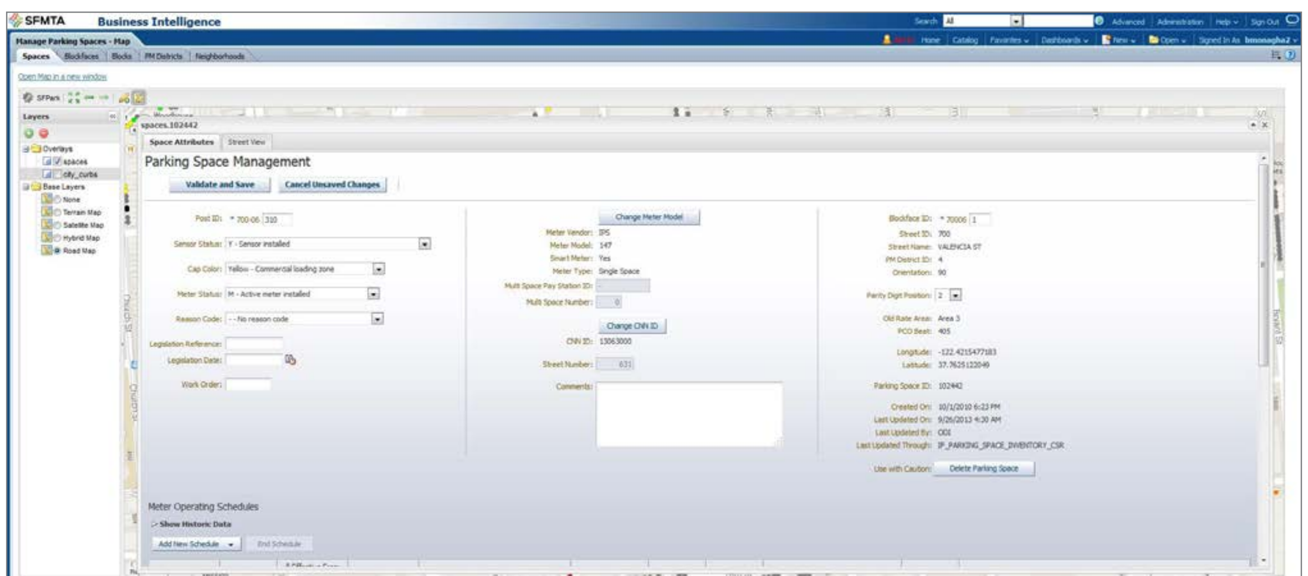
Parking space inventory detail



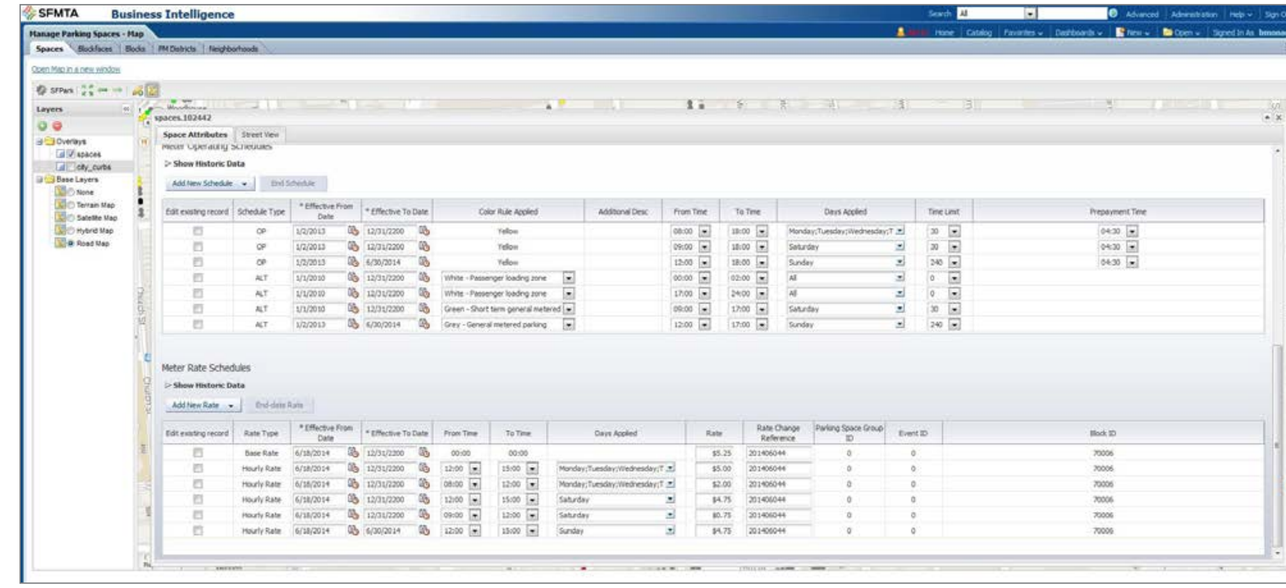
Parking space attributes



Parking space attribute editing screen



Parking space attribute editing screen, continued: Operating and rate schedules



Business intelligence and analytical tools

Transforming data into valuable information

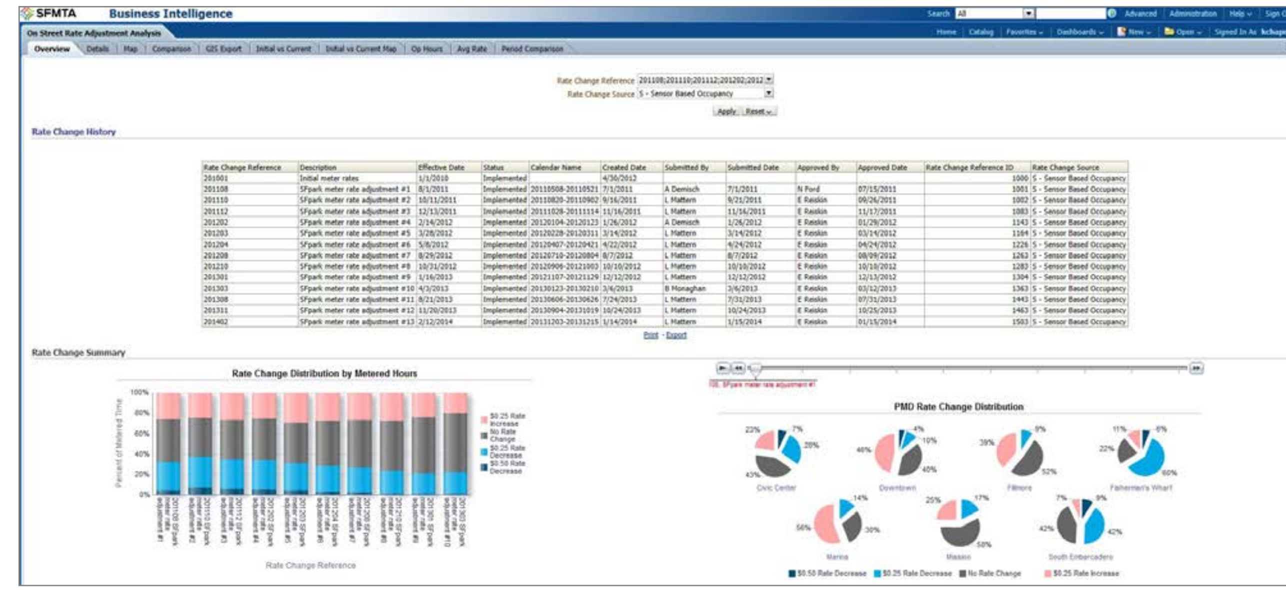
Business Intelligence (BI) reports and analytics technology allows the SF*park* system to transform enormous quantities of detailed raw data into information that is useful and can result in recommendations and actions to meet the objectives of parking management and the SMFTA more generally.

BI, in a broad sense, refers to software programs, suites, or packages which allow an organization to create standard operational reports as well as take advantage of a data warehouse to combine operational data with other data and perform analysis. This analysis can be over the dimensions of time and/or geography, as well as the oft-mentioned "slice and dice" capacity to do ad hoc reporting and research to great levels of granularity. One essential feature of the SF*park* BI system was that every night the parking space inventory details were pushed to a GIS shapefile to ensure up-to-date information necessary for analysis. This functionality may not be necessary for every project; however thinking through BI needs is important for a successful long term project.

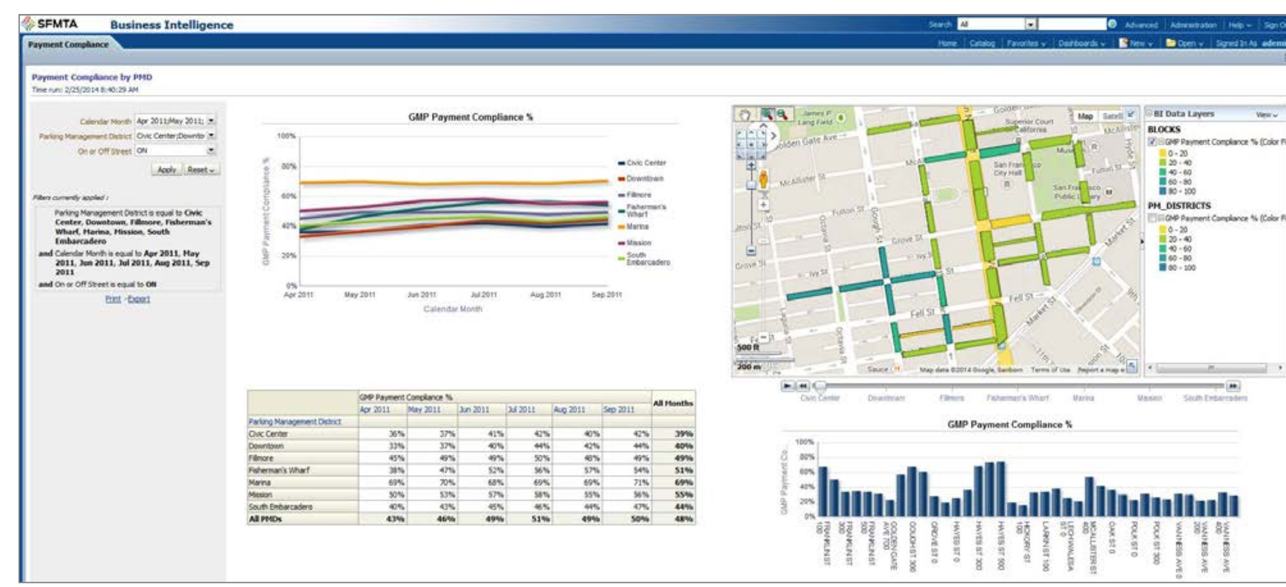
It seems likely that most agencies will use a mixture of several products. Some users will want to see and manipulate data in a spreadsheet. Some will want more complex reports already created and distributed. Others will want to take analyzed data and further process it with statistical packages and/or GIS-oriented analytical software. One common trait among BI functionality areas is that the results of all the reporting and analysis should be focused on business-defined metrics and key performance indicators. For the purposes of this document, the term BI will be used to include all this functionality.

The following screenshots illustrate BI's ability to visualize large datasets which enable the analysis of trends and performance metrics.

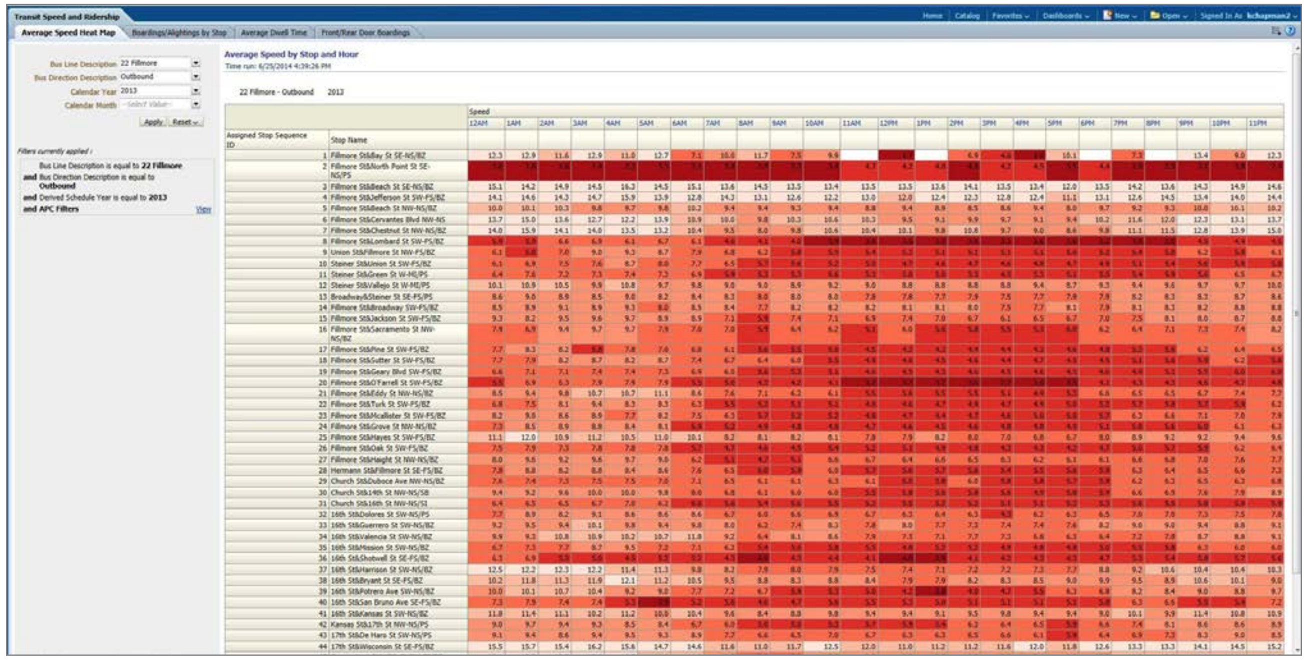
On-street rate adjustment analysis



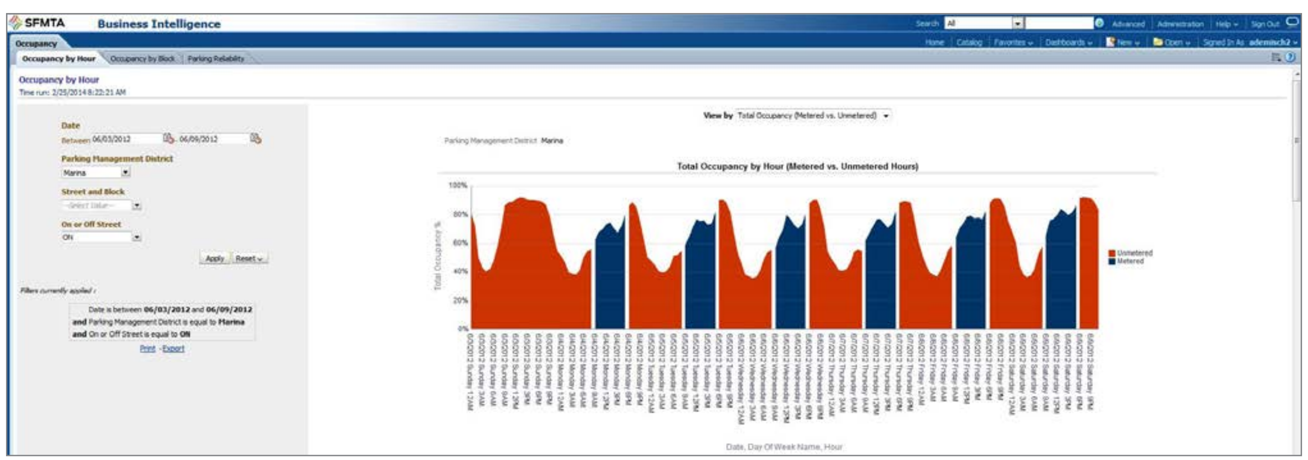
Payment compliance by parking management district



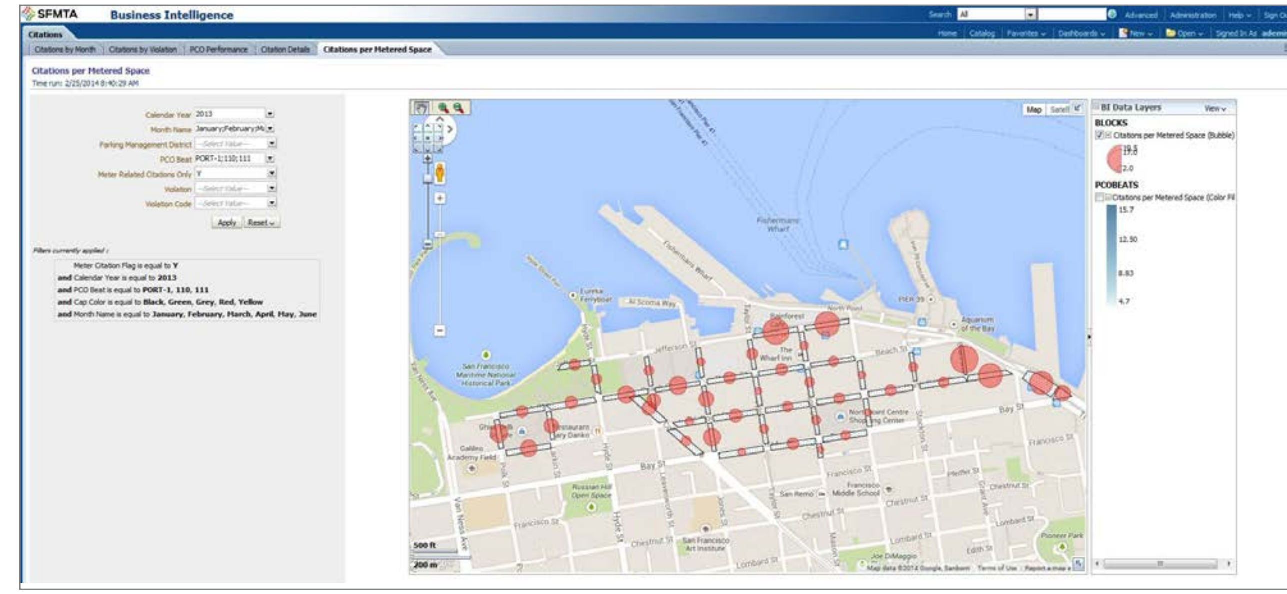
Average transit speed heatmap



On-street parking occupancy by hour



Average citations per metered space by block



Smart platform apps

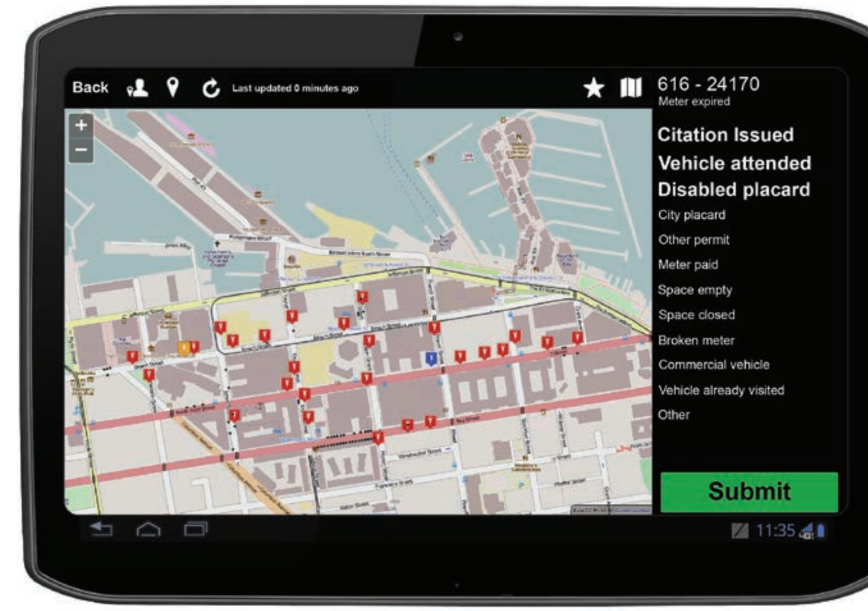
A powerful shift in technology

The burgeoning smart platforms (smartphone and tablets) and associated app business sectors are transforming the way we interact with computing.

As smart platforms become more powerful and their software matures, both phones and tablets will be deployed as both operational and analytical user platforms. BI vendors are creating apps to allow this access, and this is relevant for parking management. For example, having operational access for parking meter technicians could be valuable in areas such as work order tracking, inventory management, and even state of repair, as a technician could take a picture of the device and attach it to a work order.

In addition, asset management will be deployed using smart platforms and take advantage of the mapping capabilities to locate the asset and record information about the condition, including attaching a picture to the transaction.

Another possible application of smart platforms is enforcement. SF*park* tested an application to enable targeted enforcement called reEnforce. It is easy to envision how enforcement could be more technically directed with a more sophisticated approach. Such direction could come from real-time data if parking sensors were deployed or could be estimated by combining historical sensor data with current meter payment data. In such a case, these estimates might show where there are higher probabilities of finding parking non-compliance based on historical analysis.



The reEnforce application user interface

The SF*park* iPhone app

5.

POSSIBLE FUTURE DIRECTIONS

The SFMTA chose a specific approach to meeting the technical requirements of the SF*park* project. As other cities learn from and improve upon this example, this chapter shares high level considerations about other or new directions that other cities may choose.

Data sharing and data security

Securing shareable data

Multiple data sources are both a security challenge and an opportunity to increase data sharing and transparency.

As technology advances, cities may build semi-cloud solutions such as a city-wide communications network. Technological advances like this example require close scrutiny in order to achieve a good balance between data security and the openness of data sharing.

Data security needs to be evaluated by network type and possibly down to the network segment. For example, smart meters need some form of a PCI-compliant network to exchange credit card payment data with some banking network. That usually must be separate from any city network that would want transactional data from the meters.

Other efforts within the city may provide opportunities to partner with other agencies and share the cost of building a city-wide communications network.

Efforts at data sharing with the public must also be approached with care. Transparency is definitely a goal, but at what level? Operational data often is too voluminous to share. It also can be hard to interpret correctly. The SF*park* experience suggests that a balance can be achieved between transparency and data volume by sharing summarized data which is aggregated to a level that the user commonly thinks about, such as "by block" or "by neighborhood." Public data should be transparent and available; however, it must not be shared in an unusable form. Data that is too detailed will be unusable inasmuch as it would require subject matter expertise to explain and interpret. That role is usually not one funded within agencies.

Open source possibilities

A possible strategy to lower costs for smaller scale projects

There are many possible open source options a city or bureau can choose to employ from operating systems to storage and business intelligence. The use of open source software has the potential to dramatically reduce costs.

But it will often also reduce IT system capabilities such as highly-tested High Availability and Disaster Recovery, seamless integration of database, middleware and end applications, and already canned "bolt-ons" or adapters for interoperability.

In general, the tradeoff for the lower cost of the open source approach is usually lower reliability, robustness, and flexibility. However, it is likely that for small to medium-sized cities, open source software could be made to work suitably. This would only be possible if your developers and implementers are experienced in these open source tools before undertaking a project such as smart meter integration. Learning a new subject matter while also learning new technical platforms invariably leads to delays, restarts, and a higher risk of project failure.

Network

Agencies can set up and operate networks at their discretion. Most have standardized on a Microsoft solution, although this is not a requirement. A Linux-based network will work, as will any other Unix variant. TCP/IP is the only true requirement. Network management can be accomplished via a number of open source tools.

Although this document makes no recommendation and implies no endorsement by inclusion or exclusion from a list, below are several open source solutions.

- OpenNMS
- Zwenoss
- Nagios
- NetXMS

Server

The choice of server hardware is completely at the discretion of the jurisdiction or agency and subject to its standards.

Operating system

The operating system is completely at the discretion of the city and subject to its standards. However, care should be given to consider the differing needs of both OLTP and OLAP/BI systems and the ability of an operating system to support such endeavors. For most US agencies, Linux or Windows are viable options. Only for the largest of cities will scalability become an issue. For most other cases, the most relevant factors in a decision are existing standards, strategic technological direction, and existing technical expertise.

Storage

- There are three basic approaches to data storage. The best choice depends on existing infrastructure as well as technological direction. Cloud storage is not described, as it is beyond the scope of this document in part because it was too immature to consider in 2009 when most architectural decisions were made.
  - Locally attached
    - Cheapest
    - Easiest to implement
    - Hardest to include into high availability and disaster recovery plans
  - SAN
    - Usually the most expensive option
    - Fastest

        - Most reliable
        - Easy to include into high availability and disaster recovery plans
    - NAS
        - Cheaper than SAN, but more expensive than locally attached
        - Faster than locally attached, but usually slower than SAN
        - Typically attaches volumes to servers using iSCSI over TCP/IP
        - Can be a decent compromise
        - Sometimes transient performance slowdowns can cause database problems
- Big data and cloud considerations
    - Hadoop—more of a framework for cloud storage
    - Amazon Web Services
    - Google Cloud Platform
    - Microsoft Cloud
    - HP Cloud Storage

Database

As with the server, the operating system is completely at the discretion of the organization and subject to its standards. All of the caveats in chapter 6 of this document apply. Even more attention needs to be given to the area of existing technical expertise. In many agencies, there is little or no expertise in data warehousing or BI. Creation of a star schema can be done by consultants with the daily operations left to existing staff, perhaps with some supplemental training. It may not be necessary to have a dedicated data warehouse architect for each city department.

Middleware

Although this document makes no recommendation and implies no endorsement by inclusion or exclusion from a list, below are several open source solutions.

- JBoss
- Open Source Storage (OSS) Niazi Middleware System
- inBloom
- LinkSmart

Business intelligence

Many options exist, with some being truly open source, sometimes called "open source free products" and some being semi-open source, with charges for product support and/or extensions, and sometimes called "open source commercial products."

Examples of open source products, with some being open source free products and some being open source commercial products, are below. Please research each offering carefully when determining your BI direction.

- Palo
- Pentaho
- SpagoBI
- Eclipse BIRT (Business Intelligence Reporting Tools)
- KNIME
- Tableau Public
- SAP Lumira
- Nucleon BI Studio and BI Viewer (free versions)

None of these options have been evaluated by the project. No endorsement is intended by listing them.

Other presentation software

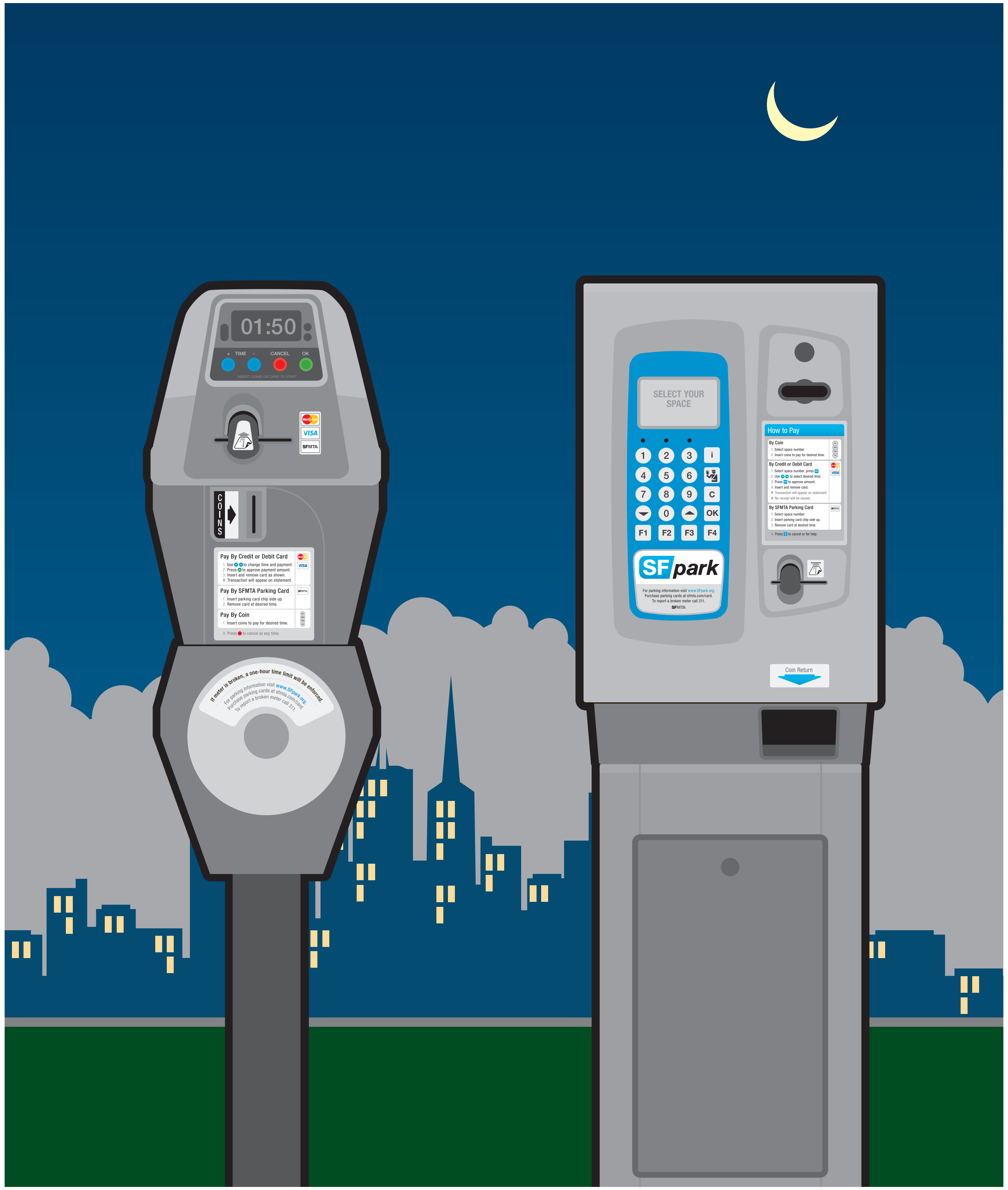
- Microsoft Excel for simpler needs
- Business Intelligence/Presentation tools such as Tableau
- Statistical Analysis packages such as SPSS and R
- GIS Analysis such as ESRI Analytics

Caveats

- With reduced price comes certain challenges
- Getting all the separate pieces to work together will be a challenge
- Training in the products may not exist
- A lot of initial work will be on-the-job training and trial and error, resulting in a system that is more a prototype than a production system.

6.
CONCLUSION

The SFMTA hopes that this guide helps other cities learn from their experience and that they can also can save time and money while improving both parking management as well as the whole transportation system by using data to make smarter decisions. The SF*park* pilot tested many new tools and technologies, through significant testing and tweaking, the SFTMA developed a system that worked for the goals of the agency. Since the inception of the SF*park* pilot, the technology and tools used have improved immensely and will become even more useful and affordable for other cities over time, enabling them to make data driven decisions to improve their quality of life.


01:50
COINS
SELECT YOUR SPACE
SFpark
How to Pay
Coin Return

SFpark.org