Photo Kiosk Architecture

Luke Stutters, Geoff Youngs, Matt Mofrad, LiveLink Technology (UK)

Abstract

The presentation 'Photo Kiosk Architecture' will briefly review the technologies that LiveLink technology has used for its photo kiosk products over the last 10 years.

The current kiosk architecture will be discussed in the context of wider industry solutions.

Finally, a novel architecture will be presented that aims to increase the flexibility of photo kiosk design.

Introduction

LiveLink Technology is a software and hardware development company based in the UK. The company has been producing various models of photo kiosk for ten years. The design of these photo kiosks has undergone significant changes over this period.

This paper discusses the hardware and software developed for LiveLink photo kiosks with an emphasis on the software. The hardware and software changes between models of photo kiosk are summarised together with the reasons for these changes.

The current hardware and software design are briefly compared to other photo kiosk solutions. This is a limited comparison since the source code for many popular photo kiosks is not open and often deliberately obfuscated.

This paper is not intended as a simple historical document - it aims to be a quantitative analysis of photo kiosk trends. To this end, technical data on the hardware and software internals is presented graphically to identify trends. On the basis of these trends, a novel kiosk architecture and a series of medium-term predictions are made. These predictions are an attempt to forecast the future of photo kiosk software development and wider industry trends.

LiveLink Photo Kiosk Hardware Evolution

This section discusses the hardware evolution of the kiosks, together with the reasons for generational changes. The aim is to establish trends in kiosk hardware over the last 10 years and the most important requirements from retailers.

2004-2006: “K series” kiosks

The first generation of LiveLink photo kiosks used a mix of modified and unmodified PCs, card readers, optical drives and touchscreens. These components were connected as a stand-alone system and installed into a number of retails locations as countertop kiosks. This hardware was chosen primarily for cost, ease of production and Linux compatibility.

2006-2008: “201" kiosks

The next generation of kiosks were built using modified all-in-one touchscreen PCs (Cybernet iOne-945s). The all-in-one PCs significantly improved the ease of production and the assembly time. These kiosks were also far more compact than the previous generation - this was popular with all retailers. During the product life cycle, it was found that the all-in-one PCs had a higher cost and lower durability than standard PC components. This made them less economic to upgrade or repair.

2009-2012: “202” kiosks

This generation of kiosks used a separate screen and small form-factor PC. Some custom parts were designed by LiveLink and produced by an external company - these parts integrated the card reader and optical drives. This was a popular and successful product with good reliability, repairability and an upgrade path for existing owners. However, the supply of these PCs were limited and many components were non-standard, leading to supply issues.

2012-2014: “505” kiosks

When suitable small form-factor PCs could no longer be sourced, a PC case with an integrated VESA screen mount was adopted. This used mostly standard internal components (eg. motherboard and heatsink) together with some custom parts designed by LiveLink and produced by external companies. This resolved nearly all component supply issues. The 505 kiosk was a popular solution with excellent reliability. However, the PC case suffered limited availability after a couple of years.

2014-: “615” kiosks

For this generation, a custom steel kiosk chassis was designed by LiveLink. This was designed using in-house CnC prototyping. This resolved supply issues and allowed standard PC components to be used throughout the kiosk. The kiosk design incorporates a mixture of 3D-printed and injection moulded components that were prototyped using LiveLink's 3D printing capabilities. Although the up-front engineering costs were considerable, the result was a significantly lowered production cost, a more compact form factor, and lower projected ownership costs over the kiosk lifetime.

We assert that this establishes the following trends:

- Kiosks are moving away from non-standard PC components towards standard components for reasons of reliability, ease of upgrade and cost
- Retailers have demanded more compact kiosks over the last 10 years
- Retailers have paradoxically wanted kiosks with larger screens. We suggest that this is due to the improved customer experience from a larger screen.

LiveLink Photo Kiosk Software Evolution

The first version of the photo kiosk software evolved from an existing project for the film industry. This was a Linux-based photo kiosk for retail, possibly the first in the world in 2004. Linux was used instead of Windows due to the expertise of the development team.

The first supported minilabs were the Noritsu QSS series. The initial Noritsu compatibility was developed without Noritsu support with considerable difficulty. However, the kiosk was subsequently adopted and promoted by Noritsu in the UK as their recommended kiosk solution.

Compatibility with other minilab manufacturers was added following customer demand. LiveLink became a more independent developer and manufacturer as a result.

The following sections outline the key software stack changes between generations.

2004 kiosk software stack:

- Debian Linux
- Ruby 1.6
- Mainly monolithic structure
- GTK modified for touchscreen use
- In-house XML-based format for complex GUI and animations
- Management using separate application in “Staff Mode"

2012 kiosk software stack (505 generation):

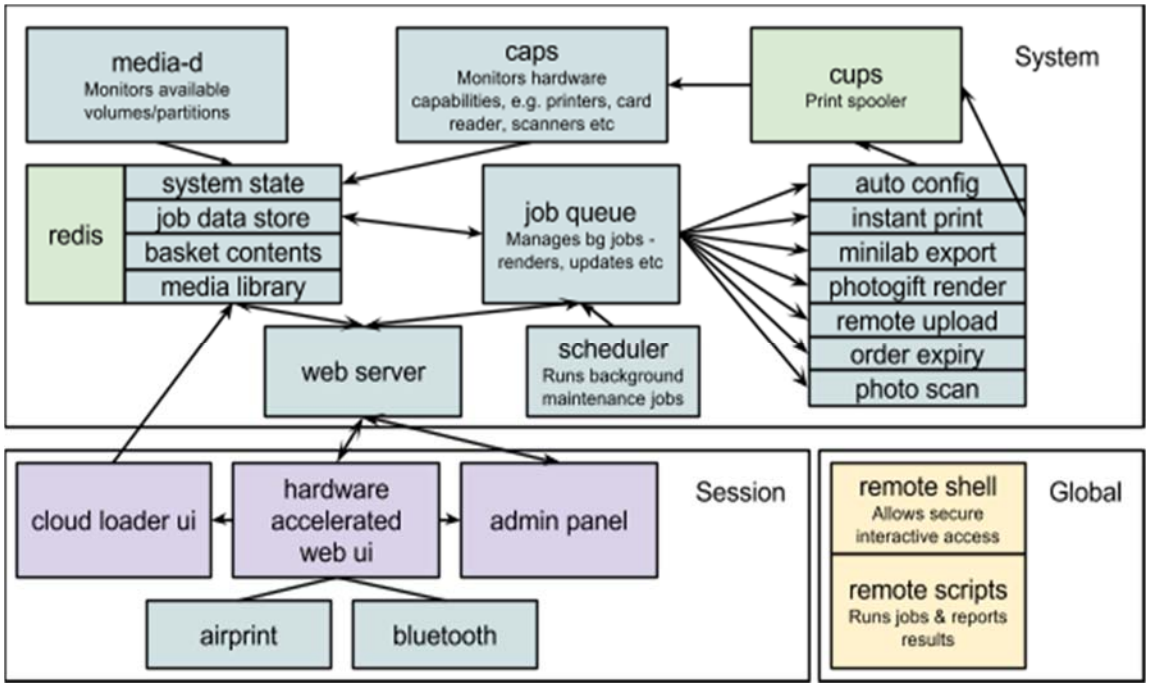
- Ubuntu Linux 10.04
- Ruby 1.8
- In-house XML replaced with web standards (HTML5)
- GTK replaced with modified WebKit for GUI to work around browser sandbox restrictions
- Separate services for:
  - Order uploads
  - Photo gift and photo book rendering
- ‘Prism’ system developed for photo gifts and photo books. This uses an extended SVG format with YAML and JSON metadata. The Prism system has since grown to over 63000 templates for 1620 physical products.
- Product management can be partially downloaded from LiveLink online photo ordering solution
- Cloud loading via direct cloud provider API (eg. Facebook Connect)

2014 kiosk software stack (615 generation):

- Lubuntu Linux 14.04
- Ruby 2.0
- Service-based architecture for gui
- Key-value database used for cart and order state
- Service-based media loading
- Cloud loading via in-house API that abstracts multiple cloud photo providers APIs
- GUI built using web standards (HTML5, jQuery, Bootstrap, SASS)
- Minilab and printer integration uses same code as the LiveLink online platform
- Kiosk management via LiveLink online platform
- 3D visualisation using WebGL

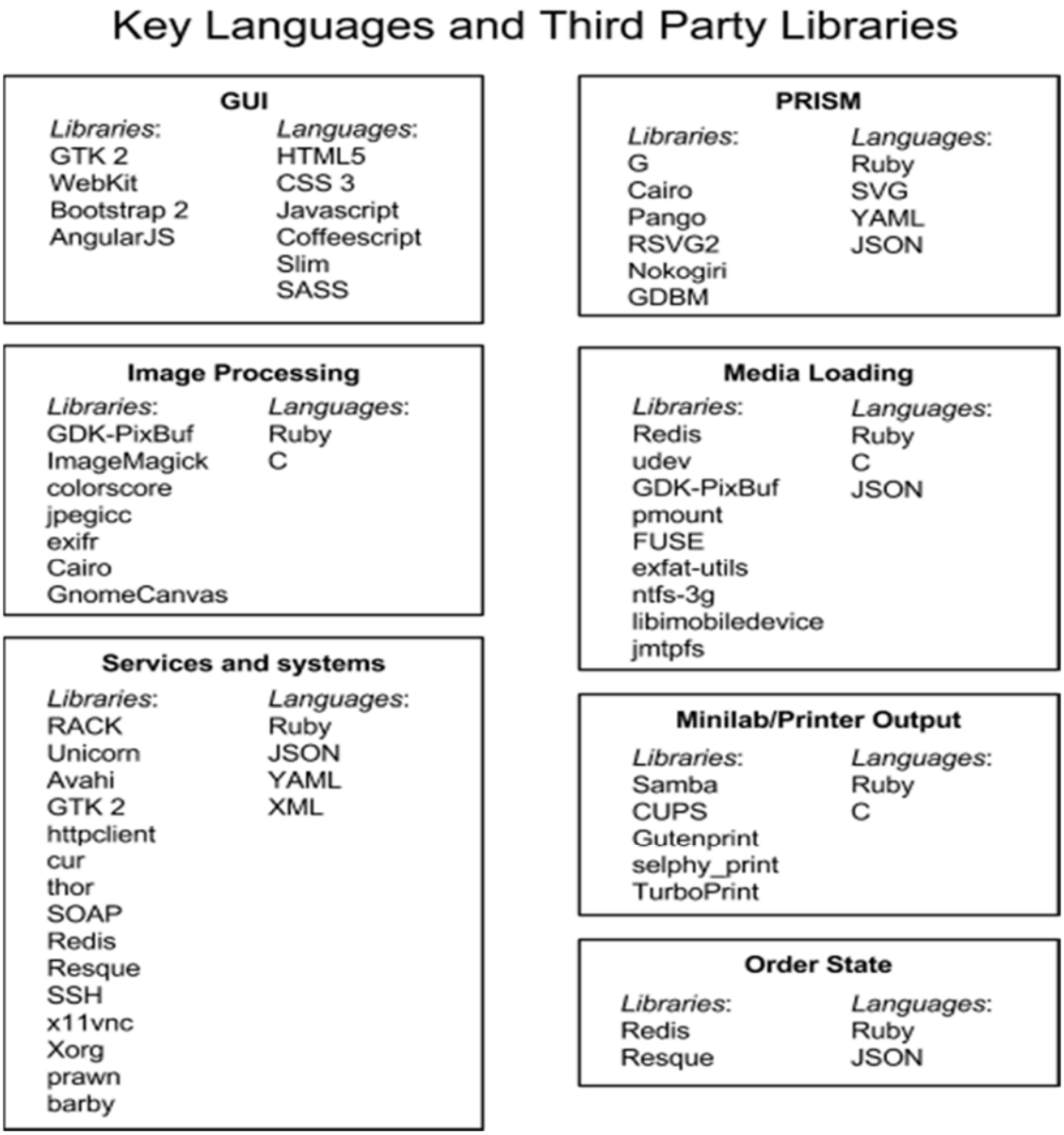
Current LiveLink Kiosk Software Architecture

In this section, we present the existing LiveLink photo kiosk software architecture in more depth. The following diagram shows the service-based architecture of the 615 generation kiosks:



Instead of a single, monolithic application, the key components are separate services that communicate via APIs and sockets. This has greatly simplified the production of different kiosks for different retailer requirements.

The key third-party libraries and languages are presented below. This demonstrates the extensive use of open-source projects:



Comparison of the LiveLink software stack with other industry solutions

This section highlights the key software differences between the LiveLink kiosk solutions and the most popular industry kiosk solutions.

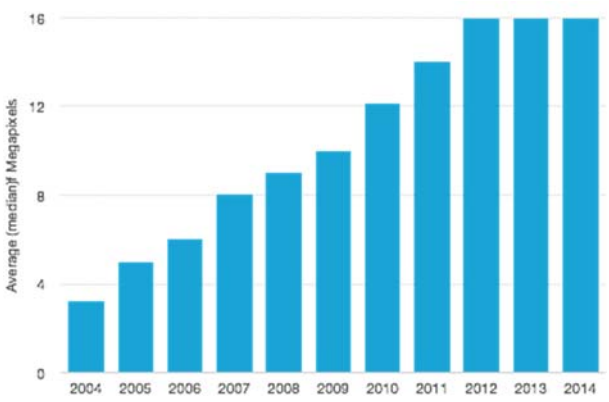
Due to the restrictive sale and licensing of most minilab manufacturer's kiosk solutions, it was not deemed appropriate to conduct an in-depth analysis of their internal structure. For this reason, the following table compares the LiveLink kiosk to current industry trends - not all of the following differences are true for specific industry kiosks.

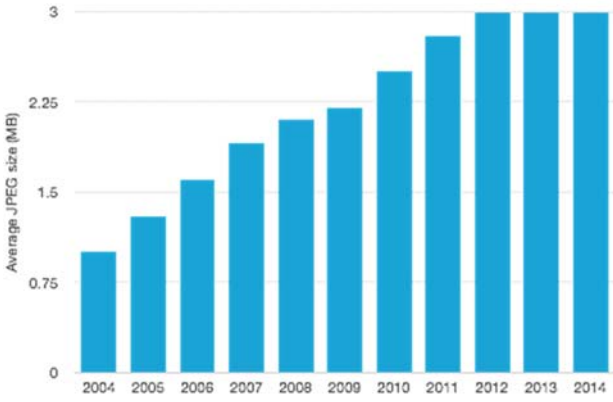
LiveLink 615 Kiosks	Industry Solutions
Linux-based	Windows XP / 7 based
GUI using cross-platform web standards	GUI using native widgets and libraries (eg .NET)
Management through online platform	Limited integration with online solution
Extensive code shared with LiveLink online platform	Minimal shared code with online platform
Interpreted code that uses mainly open-source libraries	Compiled code that uses closed source libraries
Android and iOS media loading through open source libraries and protocols	Android and iOS media loading through closed source libraries or apps

LiveLink Kiosk Software and Hardware trends

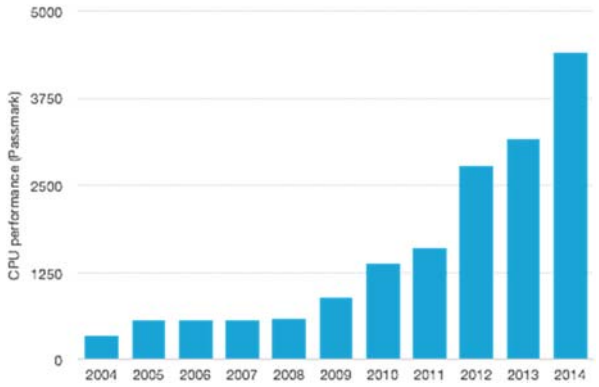
In the process of developing our proposed kiosk architecture, the software and hardware of LiveLink photo kiosks over the last 10 years was analysed. We present the results for the first time in this paper:

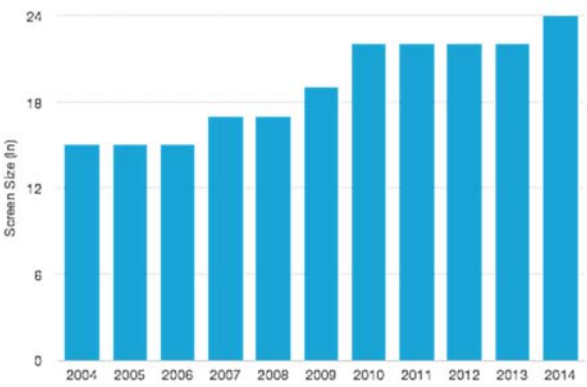
Year	Average Megapixels of customer images	Average JPEG size (MB)	Hardware Performance of Kiosks (Passmark score)	Number of Kiosk Executables	Screen Size (inches)	Lines of Code	Library dependencies (internal and external)
2004	3.2	1	340		15		
2005	5	1.3	575		15		
2006	6	1.6	575	58	15	44357	248
2007	8	1.9	575	73	17	55030	295
2008	9	2.1	594	89	17	82461	380
2009	10	2.2	889	94	19	98077	433
2010	12.1	2.5	1376	100	21	106865	452
2011	14	2.8	1600	101	21	119627	472
2012	16	3	2772	136	21	159197	532
2013	16	3	3173	173	21	176074	643
2014	16	3	4405	210	24	232381	700





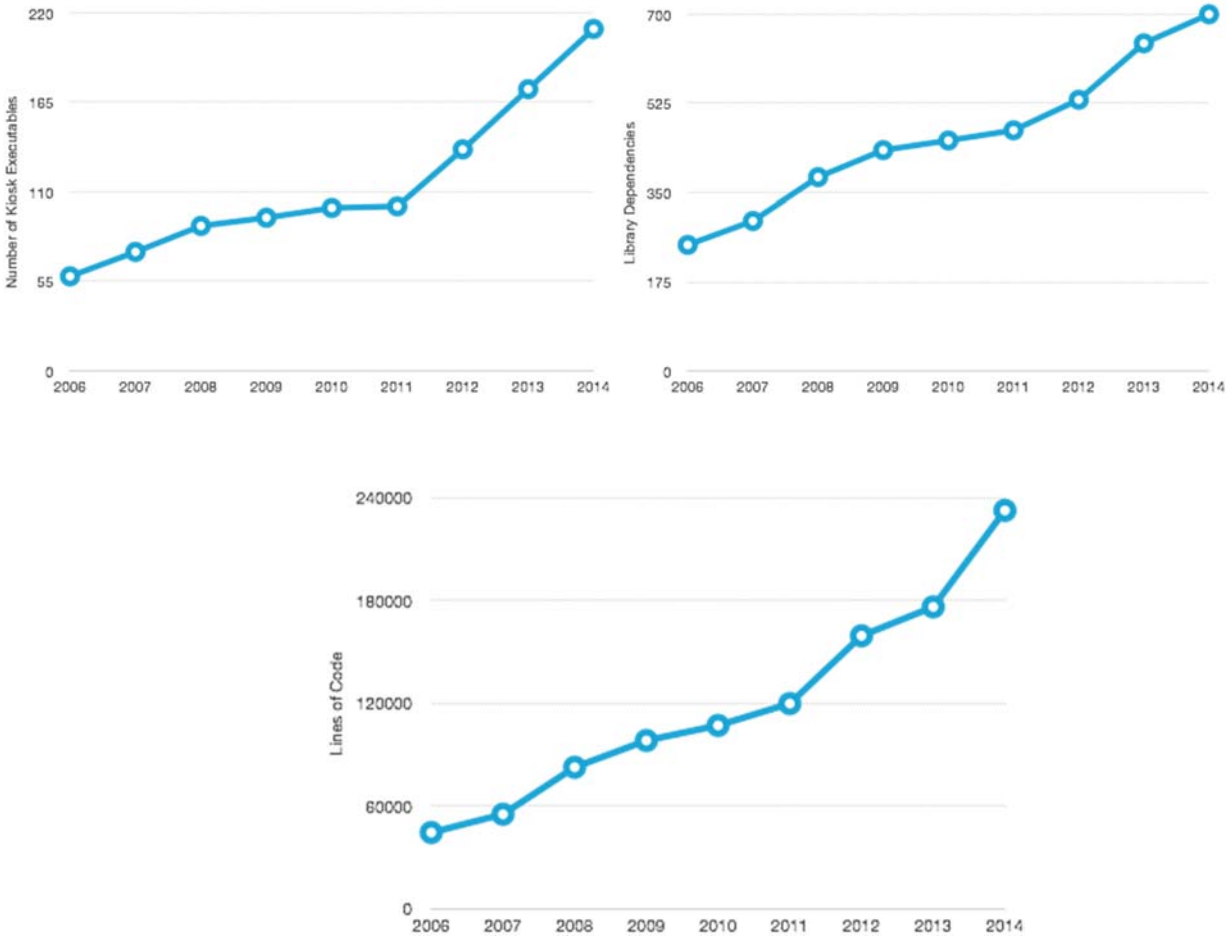
The megapixel and JPEG size charts show that although the average megapixel size of images has increased, this is a linear trend. The file sizes from customer images are, in reality, significantly larger than the 1-3MB in the chart - customer images are often around 10MB. However, the same linear trend is observed. We propose that the storage and processing requirements of kiosks are not likely to significantly increase in the medium term..





The above charts show the CPU performance trends and kiosk screen sizes over the last 10 years. The kiosk software does not have significant CPU requirements so we selected the highest value processors at the time of manufacture. We propose that there is increasing scope to optimise the CPU usage by serving multiple customers from the same PC.

The touchscreen size has also steadily increased. This reflects the availability of larger touchscreens and the demands of retailers for the improved customer experience that a large screen offers. We believe this demonstrates that retailers are unlikely to want a kiosk with a smaller screen unless there are significant associated benefits.



The above charts show the increase in the number of separate executables, library dependencies and lines of code in the LiveLink kiosk software since 2004. Again, the trend is approximately linear. The projects' growth is due to a combination of new features, the need to support new media sources and kiosk hardware changes.

As the kiosk codebase grows larger, it becomes more expensive to maintain, despite significant advantages in development tools over the last 10 years. We believe the above charts are representative of other industry kiosk solutions, although there is no public data.

We propose that the cost of developing and maintaining a photo kiosk solution has increased and this will continue to increase unless significant changes are made to the development process of photo kiosks.

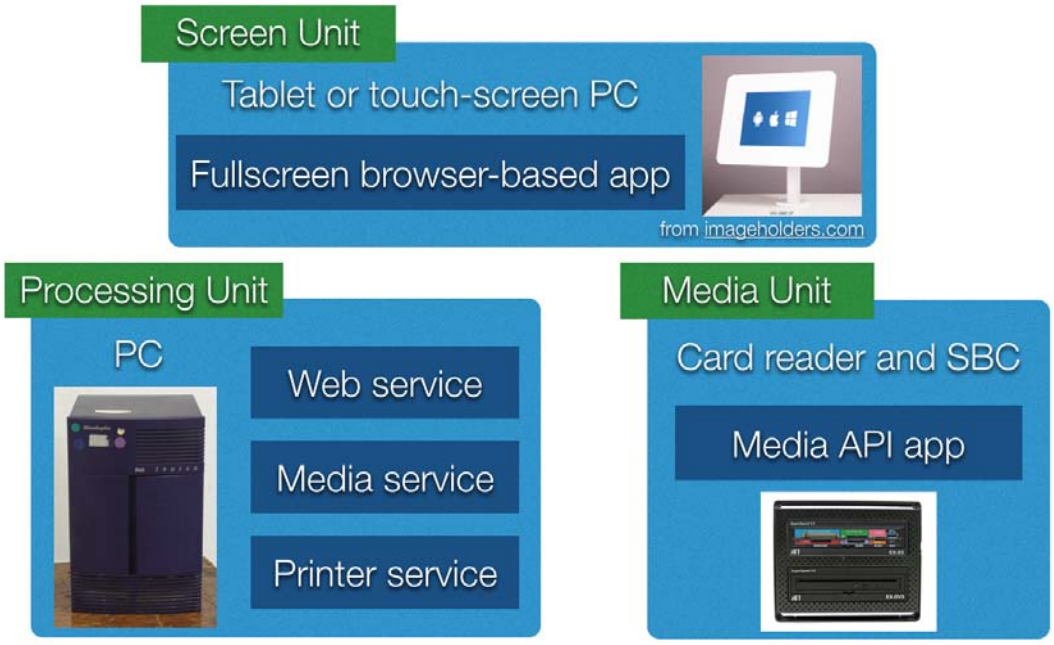
Proposed Photo Kiosk Architecture

On the basis of the above trends, we propose the following photo kiosk architecture. The core of the proposal is to split the photo kiosk into three separate components attached by a local network. This differs from existing kiosks where the screen and media readers are invariably attached to the same PC which also runs the kiosk software.

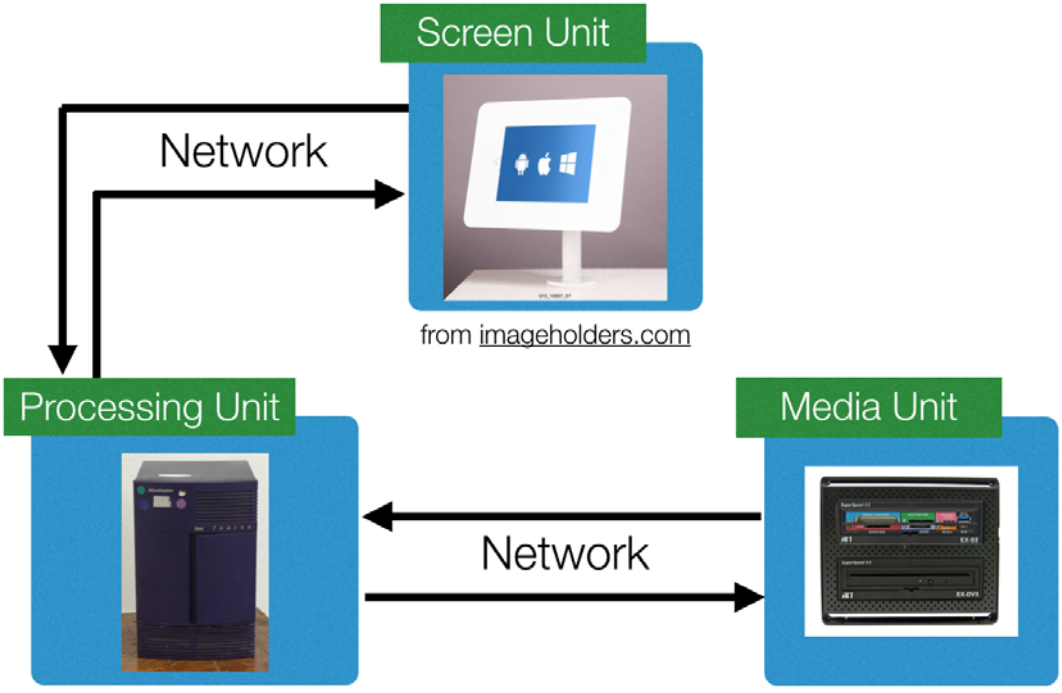
The three separate components are:

- Media Unit - optical drive and/or card reader attached to an affordable ARM-based Single Board Computer. Discoverable and accessible over network by processing unit
- Screen Unit - PC or android tablet running simple full-screen WebKit app (displaying pages served by processing unit). This app simply displays the web pages that are served by the Processing Unit. Android firmware may be modified to restrict access to other tablet functions.
- Processing Unit - PC that serves web pages to the Screen Units and reads photos from Media Units. Also runs core kiosk services, eg. output to minilab (or instant) printers and order uploads

The following diagram illustrates the proposed responsibilities of each unit:



This diagram illustrates the proposed network connectivity:



The above design does not prevent all three units being housed in the same counter-top unit like a conventional photo kiosk. However, it introduces the possibility of a more avant-garde "iPad on a stick" installation for a more modern retail experience.

We suggest the following advantages of being able to physically separate the units:

- More in-store placement options - a Screen Unit tablet occupies a fraction of the space of a conventional kiosk

- Independent media reader placement - can be incorporated into furniture or omitted to make a wireless / social media / cloud kiosk
- Fewer in-store PCs - multiple screen units can be powered by a single Processing Unit

The key software development advantages of this proposed architecture are:

- Immediate separation of concerns between each unit
- Smaller, largely independent projects
- Can use existing codebase with minimal changes
- Minimal expensive mobile development - tablet hardware can be used and changed without extensive rewriting
- User interface uses web standards which are supported in every modern mobile browser, including iPad and Android tablets
- In-store wifi could potentially turn any customer's tablet into a kiosk without needing to install any software

The key hardware advantages are:

- Reduced dependency on hardware manufacturers or form factors
- Simpler design and integration of kiosks into store furniture
- Reduced production costs
- Modular upgrades and replacements

The key disadvantages of the design are:

- More things to go wrong
- Kiosk function entirely dependent on the network
- Significant change from existing kiosk designs
- Developers may need a wider range of skills
- Novel and therefore unproven

Conclusion

In this paper, we have described evolution of hardware and software of LiveLink photo kiosks over 10 years of development. The key technologies used by the current generation of LiveLink photo kiosk software have been summarised. We have made a limited comparison with most popular photo kiosk architectures and proposed a novel architecture based on the software and hardware trends presented in this paper.