mayelespino.github.io

Mayel Espino's Resume

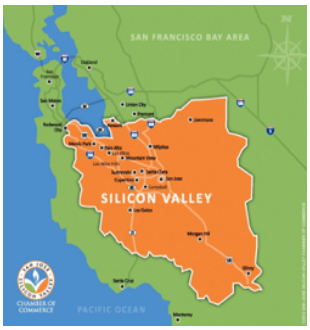
Mobile: (510) 512-5323

email: job@mayel.info
or: mayel.espino@gmail.com

LinkedIn

Download PDF files:
LinkedIn Profile
This resume

Area I am willing to commute to:



Highlights

- Received award for Microsoft patent submission.
- Received 2 applause awards at Symantec
- Inventor of 6 patents: 8,849,892 - 8,307,378 - 8,140,980 - 8,015,574 - 7,783,593 - 7,243,355
- co-inventor of 1 patent: 9,661,142
- My team won the first place in CQES first "hack-a-thon" at PayPal
- Received the "PayPalian" award for the Airlines Project

Skill Set

Programming languanges

C	20 years	Update and maintained homegrown messaging middleware solution which supports 12 different operating systems, 120 applications and 350K messages. Wrote a driver to control

		tape drive for Fedex vendors to ship files, to process thousandsof files daily.
C++	16 years	Added 4 mayor features to credit card processing (back-end) servers and also process massive transaction files. One mayor feature was the airline ticket processing for PayPal.
Python	06 years	Wrote command line utility to drive the inhouse deployment system, and used it to drive the end to end tests. The deployment system deployed 50 to 120 packates to >10K machines. Triage and fix bugs for a mobile security system that involved: Messaging/middleware systems, Monitoring tools, REST services.
Java	02 years	Created wab application using servlets to run and monitor Maven test cases.
Assembler	01 years	Wrote embedded systems in college. Extended a propreatary (SSPL) language to add methods for writing files.
Ruby	06 months	Developed puppet modules, Developed Chef recipes and tests on mixed cloud and bare metal farms or hosts.
GO	Learning	Have completed training classes, Completed a one day training on TiniGO (IoT).
C#	Learning	Implementin a set of C# microservices (Azure Functions) arround a graph database. This system models processes, assets, teams and their complex relationships.

Infrastructure

Configuration management	04 years	Developed and improved on a variety of tools, both in house and off the shelf, to enable the Release Engineering team deploy, configure and manage the entire company’s software stack.
Cloud computing	3 years	Provisioned and managed nodes in both Azure, AWS and OpenStack.
Oncall	06	

support	years	Provided second level support and tools
CD/CI	04 years	Implemented various CD/CI pipelines using: Chef, Salt and Ansible.
SQL DBs	07 years	Schema design ,performance tuning, SQL, PSQL, API development.

Operating Systems

Linux	14 years	Systems Administration and management, Process and service monitoring.
Unix	11 years	Systems Administration and management, Process and service monitoring, Ported C/C++ code on to various variations of Unix.
MVS	04 years	Wrote applications C, REX, and proprietary languages.
Windows	10 years	Software Development in C++, Java and C#.

Web

PHP	01 year	Developed in-house applications for test tool.
JavaScript	learning	Developed Service monitor using JavaScript.

Other

MQ Series & WebSphere	01 year	Developed code for complex messaging and message transformation applications.
Tuxedo	01 year	Developed servers using tuxedo’s API.
Project lead and management	07 years	Lead large teams, up to 9 engineers, in design and project tracking meetings. Day to day mentoring and project tracking.
		Maintained and extended in house infrastructure, worked on

Internet security	2 years	tooling for security companies with 1K-2K hosts. Developed a multy threaded, file processing tool to read, analyze and update the security incident databases for 10K+ files a day.

Experience

Microsoft	**Microsoft**	**February 2018 to Present**

`Fleet management – Process engineering` : This team was created to find solve gaps in: Our processes and tools used to manage/track Azure's hardware assets around the world. My responsibilities included:

- **Research, documentation and mapping of existing processes and data flows**
- **Designed and implementing solutions for**:
- **Halcyon** A graphdb to map the complex relationships between: hardware assets, data producers, data consumers, processes and tools.
- **Event bus**: An interface for fleet tools to publish key events and for tools to subscribe to the topics so that they can be triggered and process key events in real time.
- **Reconciliation engine**: A scalable, real time framework to validate that what is handed over and what is received match. For example: When we receive an hardware order, reconcile what is received at the docks with what was ordered.
- **Validation engine** : A scalable, real time framework to check that the sequence of events being published occur in the correct sequence. To measure and quantify the gaps and measure the progress of the changes we make.
- **Stood up infrastructure for Trix** : Created the production and development environments, which included AKS clusters, docker registries, a traffic manager profile, keyvautls, etcetera.

`Fleet management` : During the first months in the team, I my responsibilities included:

- **Deprecating Core XT for DND and onboarding to Onebranch**: Onebranch is the internal build and deployment pipeline that is gaining support across the entire company. CoreXT is a cumbersome legacy framework which I replaced in favor of the new standard .NET libraries and framework which is used outside the company and have much better support.

- **New hire exercise**: The best way to onboard a new hire is going through the same process and using the same tools their teammates use day to day. I created a new hire exercise to implement this for new fleet engineers.

	**Apple**	**April 2016 to December 2017**

`Software engineer – AML Infrastructure Engineering` : As part of the team I helped managed and maintain a large infrastructure to serve field diagnostic applications for Apple world wide. Highly available, multi-site. My responsabilites included:

- **Release Management** Deployed code to production and non-production environments. Help coordinate the use of the non-production environments with multiple development teams.
- **Triage** Managed ticket queue. Assigned tickets to developers and other team members. Help Development team in debugging and root cause analysis.
- **Crisis Management** Participated in the on-call rotation for High priority tickets tickets.

`saltstack – python – bash scripting – Ansible – Cassandra – Hadoop`

	**FireEye**	**October 2015 to April 2016**

`FELabs Infrastructure DevOps Team:` R3esposible for: Development of automated tools for monitoring, management and deployment of horizontaly-scaled, highly-available cloud systems. Architecting multi-datacenter monitoring facilities. Building performance and reliability metrics in a dynamic application environment. Integrate with engineering teams to provide expertise and requirements.

`Ansible Tower, Prometheus, ELK Stack, Docker, Python.`

	**Ooyala**	**March 2015 to August 2015**

`Site Reliability Engineering (and Infrastructure) Team:` **Research** and selecting or extending the right frameworks, tools and technologies for Ooyala's core platform. **Continuously improving** and re-engineering the platform architecture to handle our rapidly growing traffic. Designing and implementing clean APIs and protocols for other teams to

build features on top of. **Troubleshooting** and resolving critical issues with the platform. **Guide and support** product engineering teams in making full use of the platforms and tools the team provides

`Chef, Ruby, AWS, MetaCloud, Consul, Data Dog, Haproxy, Squid, Jira`

	**Symantec**	**May 2014 to March 2015**

`Mobile Security Customer Response Team:` **Investigate and resolve** customer issues with Symantec's Mobility Manager server (Mobile security). **Held online meetings with customers** to help them resolve difficult issues. **Designed and implemented** a "HotFix" process for quick delivery of code fixes or patches to customers. **Developed** "Analog", an application log analysis tool. Created a SymExchange space for CRT support

`Python, Django, PyCharm, SVN, Vagrant, AWS, Monit, celery, Rabitt MQ, messaging.`

	**PayPal**	**October 2006 to May 2014**

`Infrastructure Team:` **Developed** infrastructure for and drove pilot program for the Internet Security team, based on IBM's AppScan.

Developed infrastructure for testing messaging infrastructure, called AMQ.

`Stage Reliability Tiger Team:` Hand picked for **Stage Reliability tiger team**, to tackle: "the most critical issue affecting productivity at PayPal. **Designed and developed** a monitoring tool called "Argus". Which monitors PayPal Services, restarts them and sends metrics data to a backend DB for reporting and analysis.

`Deployment Infrastructure` : **Design and developed** command line interface client/driver for PayPal's Dispatcher. A Deployment System, built on top of Puppet.

`FinProd:` **Developed** for the high visibility integration of Bill me latter to the PayPal website.

`FinSys:` **Project managed and developer** code for several large scale, credit card

payment processing projects: FDMS integration, AMEX, and Airlines.

`C++, Java, Perl, Python, Ruby, Bash, AppScan (IBM), Jenkins, Jaws, Maven, github, RH Linux, REST API, Oracle DB, Django, Puppet,SOAP, clear case, RH Linux. Project Management.`

	E*trade	**February 2006 to September 2006**	C++, Tuxedo, Java, Perl & SQL, Linux.
	Kabira	**August 2005 to January 2006**	Kabira's proprietary language C++, Solaris Unix.
	Verizon	**1996 to 2004**	C++, IBM' Unix (AIX), HP's Unix, Solaris, MVS, Windows and many others. Web-sphere, MQ series, Java Servlets, Insure++, xdbc.
	Candle Corporation	**1991 to 1996**	Candle's proprietary languages, C++, Rexx, assembler. Project Management.

Education

	U.A.C.J.	1984 - 1990	B.S. degree in computer engineering, fully accredited in the United States.	U.A.C.J
	Colorado Technical University	1998 & 1999	C++ and Object Oriented certifications.	Colorado Tech.

Personal projects

- Code examples. - Some code exercises i've done.
- Picron. - This is my REST API driven alarm clock and home alarm. The code is here.

- Home Monitoring system. - This project is comprised of two raspberry pies, one web server and one to manage the sensors. The code is here
- My personal page