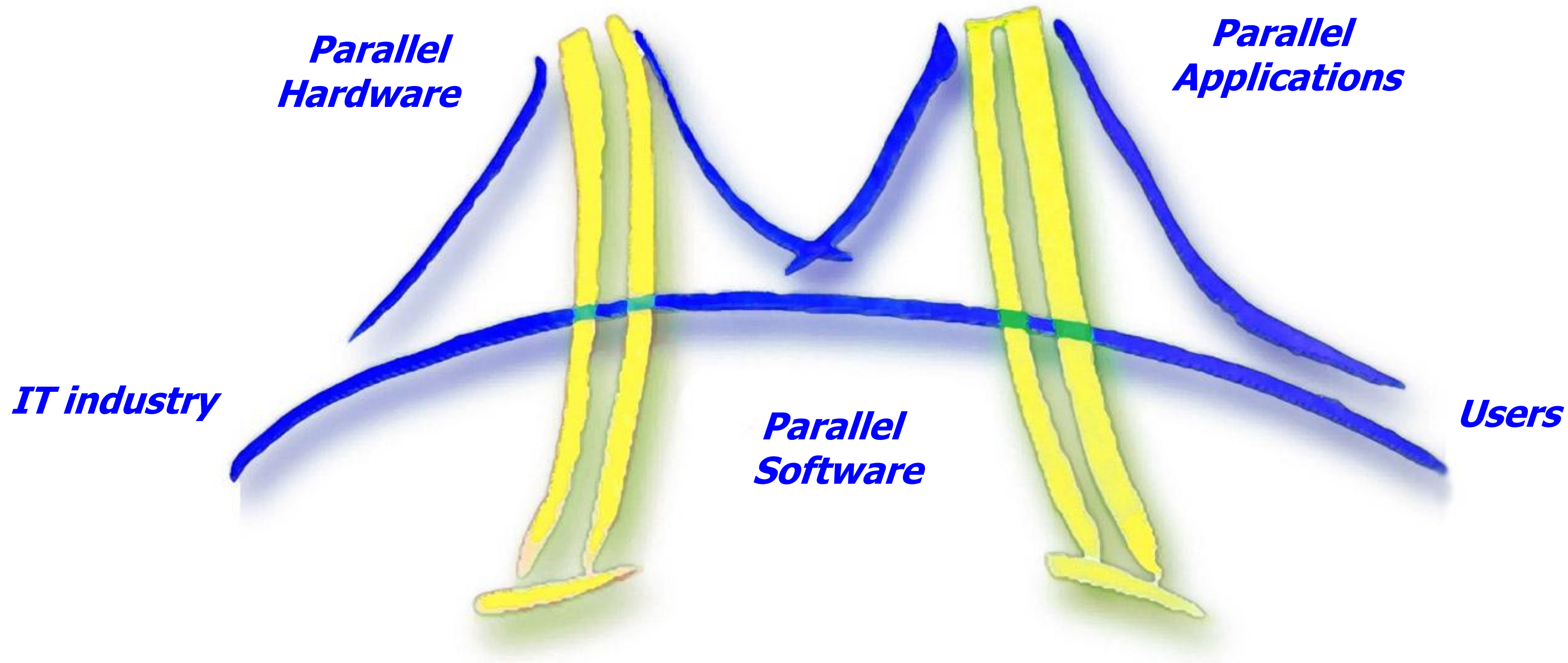
The ParLab Stack

Parallel Computing Laboratory

Sarah Bird

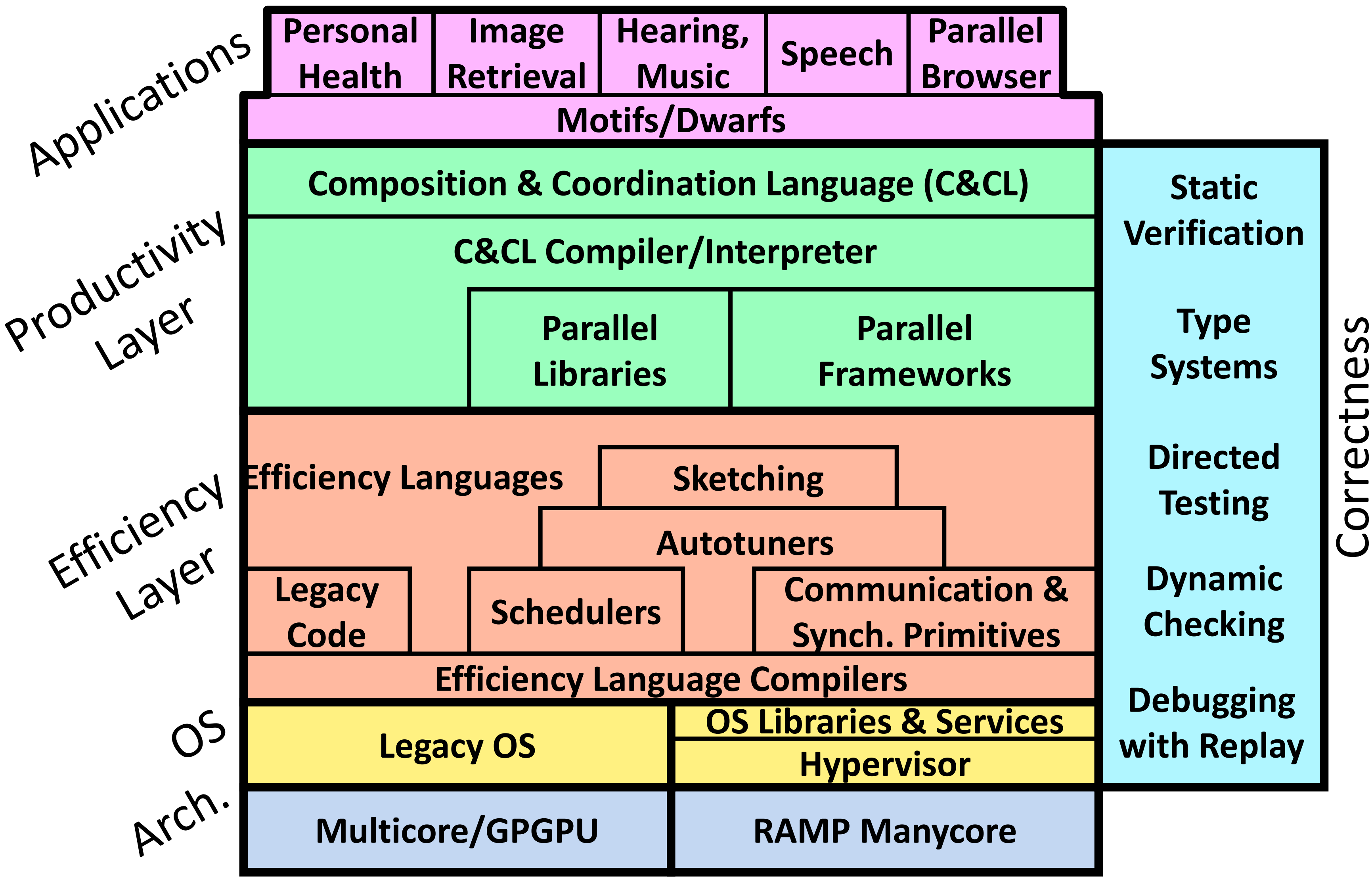
May 30, 2013

Bridging the Gap



Easy to write correct programs that run efficiently on manycore

Integrated Software Stack



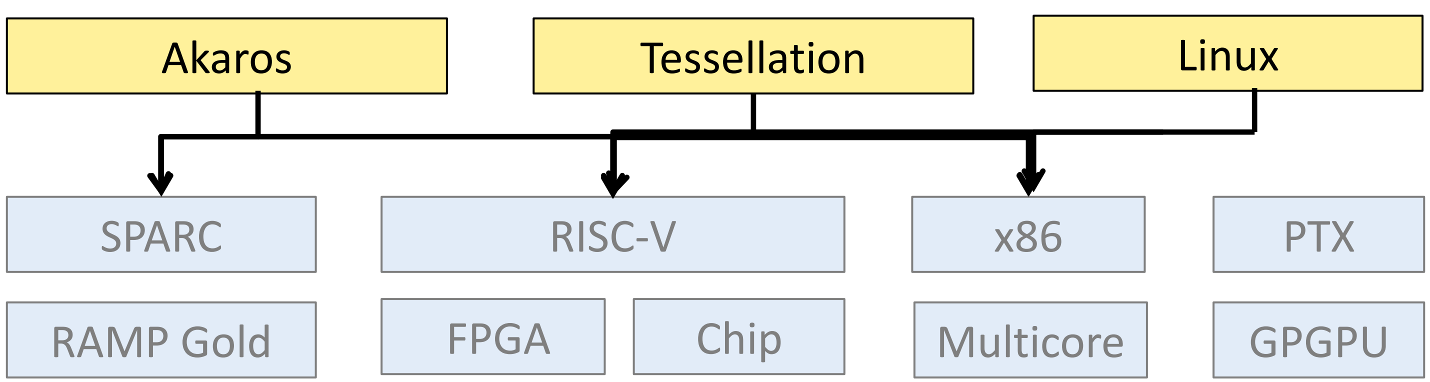
Hardware

- RAMP Gold
  - Simulation
  - 64 cores
  - FPGA
- RISC-V
  - Implementations written in Chisel
  - Rocket
    - 6 stage in-order
  - Hwacha
    - 64 bit vector core
  - FPGA
    - 2 Rocket
  - 45 nm Chip
    - 1 Rocket, 1 Hwacha
    - 1 Ghz
- x86
- GPGPU
  - Cuda
  - OpenGL

SPARC	RISC-V		x86	PTX
RAMP Gold	FPGA	Chip	Multicore	GPGPU

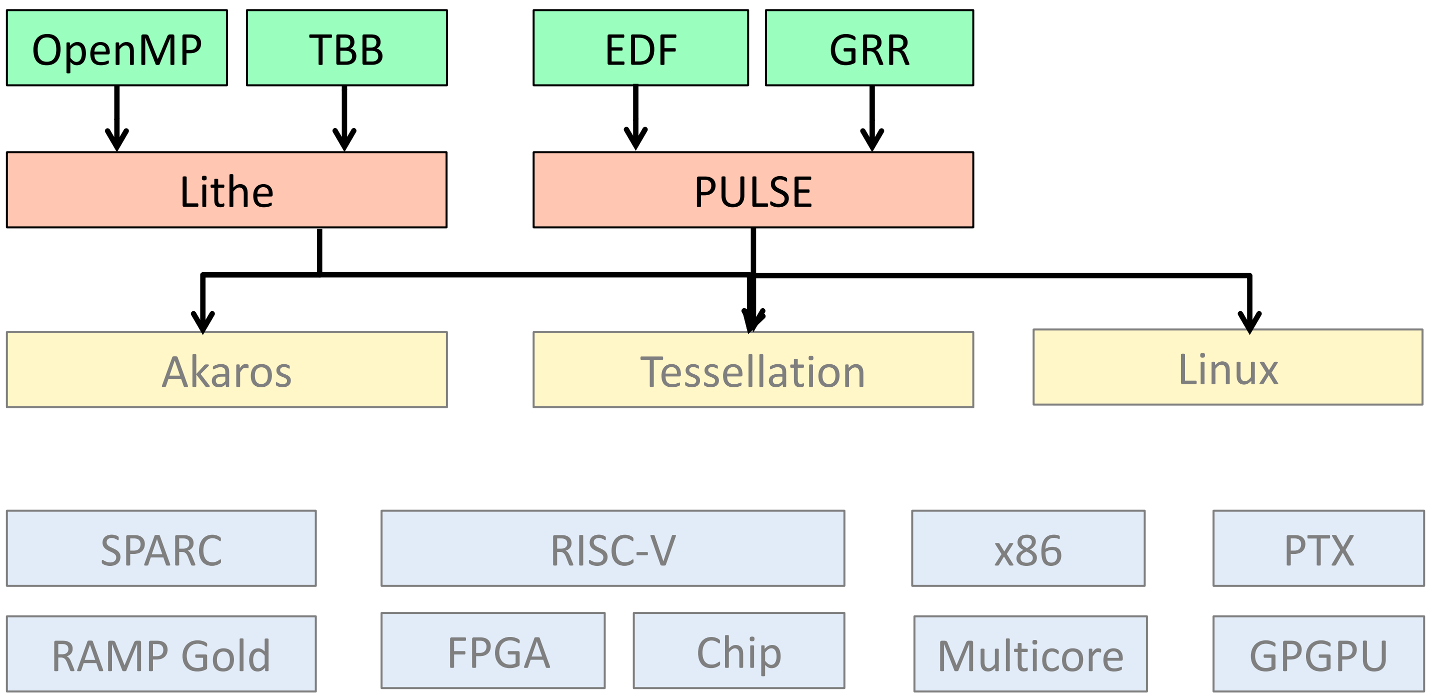
Operating Systems

- Akaros
  - Cloud OS
- Tessellation
  - Client OS
  - Space-Time Partitioning
  - Two-Level Scheduling
  - QoS to Applications
  - PACORA
- Linux

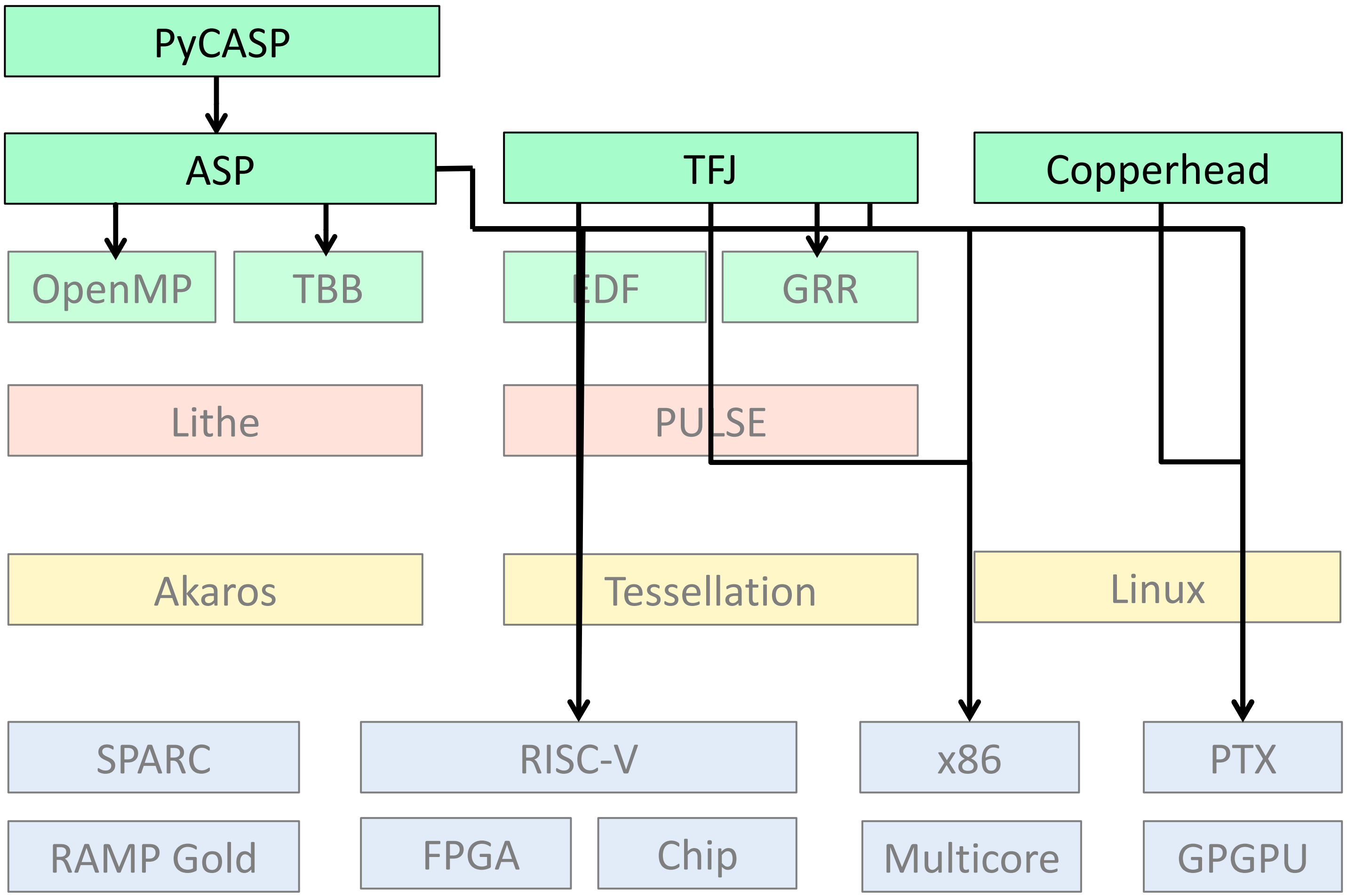


Schedulers

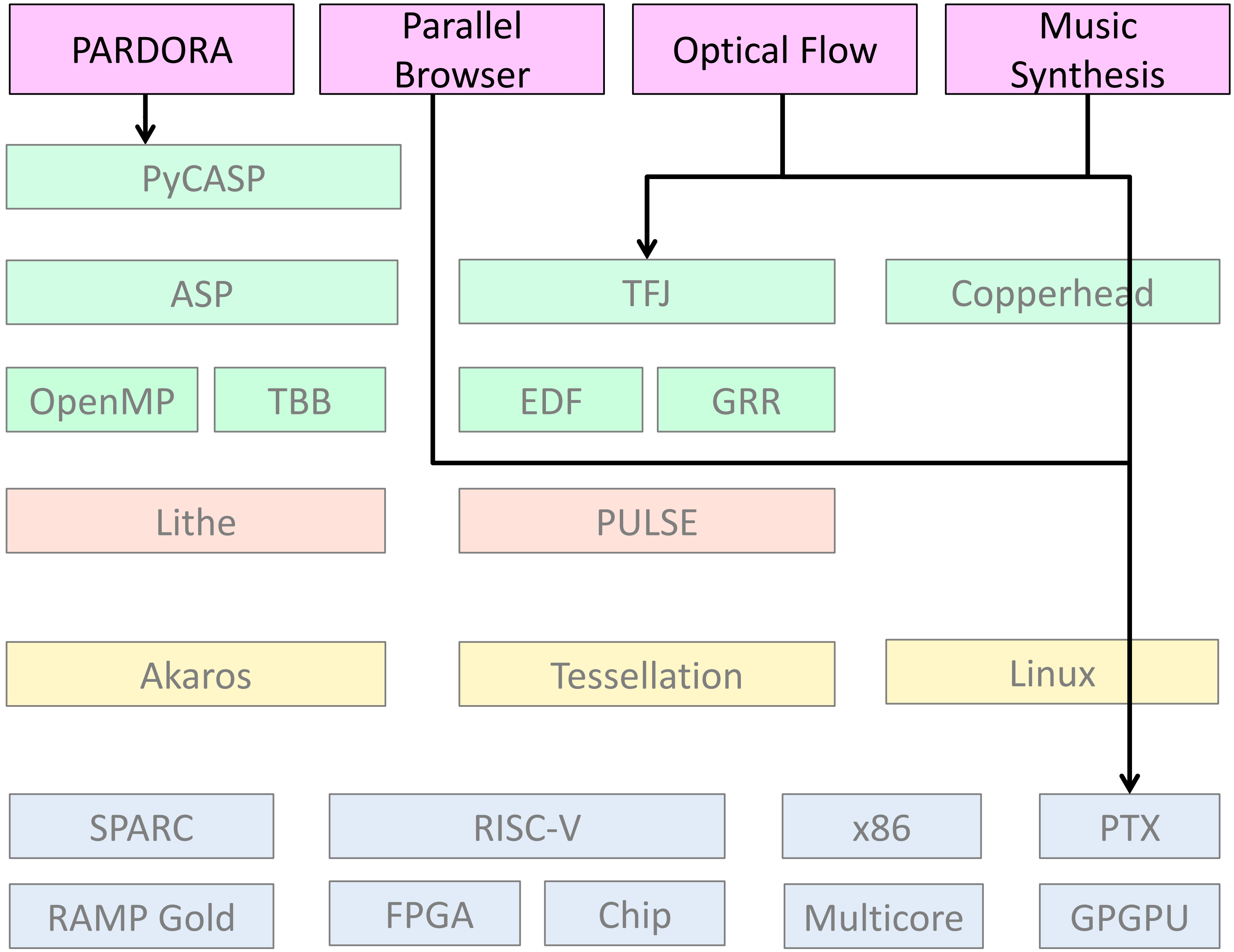
- Lithe
  - Compose Parallel Runtimes
  - Thread Building Blocks
  - Open MP
- PULSE
  - Framework to write schedulers
  - Earliest Deadline-First
  - Global Round Robin



SEJITS


PyCASP
ASP
TFJ
Copperhead
OpenMP
TBB
EDF
GRR
Lithe
PULSE
Akaros
Tessellation
Linux
SPARC
RISC-V
x86
PTX
RAMP Gold
FPGA
Chip
Multicore
GPGPU

Applications



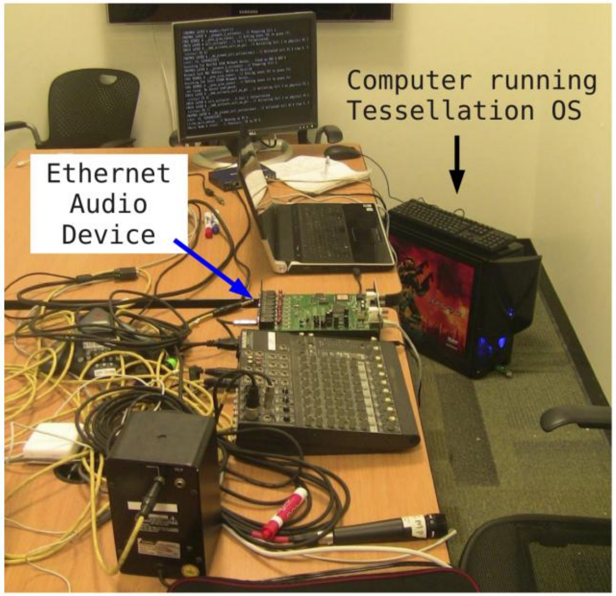
Why Create an Integrated Prototype?

- Encourages Collaboration
- Prevents neglecting important pieces of the problem
- Uncover opportunities for invention by seeing which side of an interface is the best place to satisfy a requirement

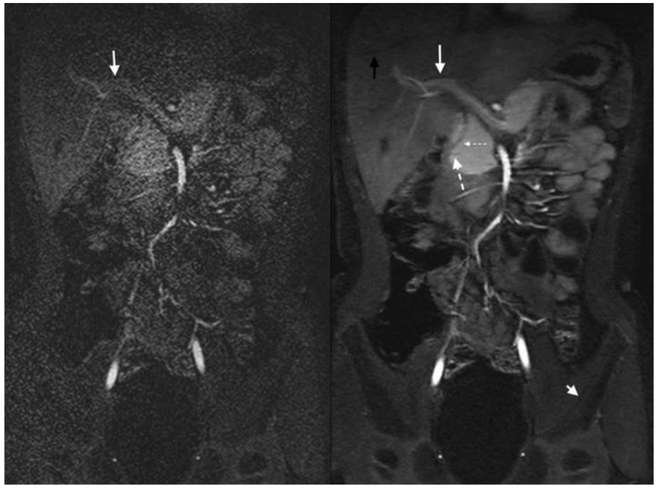
- Demonstrate the importance of design simplicity
- Enhance the education of the PhD students in areas beyond their own specialties
- Help with technology transfer by giving concrete examples of our ideas for our colleagues in industry

Forces for Integration

- Design Compatibility
  - Shared Space and Discussions
  - Symbiotic Designs
  - Example: Music and Tessellation
- Customized Support
  - In-house experts helping adapt their design to your problem
  - Examples: Lithe and Tessellation
    CAA and Applications



Preparing the Music Demo on Tessellation OS
for the Winter 2011 Par Lab Retreat.



- Motivating Applications
  - Exciting to show your research on run a compelling application
  - Examples: Patterns and MRI
    BFS and RISC-V

Integrated Demos in ParLab History

1 January 2010		2 January 2011	3 January 2011	4 January 2011	5 January 2011	6 May 2011		7 January 2012	8 January 2012	9 January 2012	10 January 2012	11 January 2012	12 January 2012	13 May 2012		14 May 2012	15 May 2012	16 January 2013	17 January 2013	18 January 2013	19 January 2013		20 January 2013	21 May 2013	22 May 2013		23 May 2013	24 May 2013	25 May 2013	26 May 2013			27 May 2013		
1	1	2		4																	19			21	22	22				26	26	26			
			3			6	6	7											17			19					23								
											10						15	16					20					24					27	27	27
																				18															
					5				8				12			14	15												25	26	26	26	27	27	27
	1					6	6		8	9				13	13	14			17		19	19									26	26	27	27	27
																									22	22				26					
						6	6		8	9				13	13	14			17																
										9																									
						6	6		8					13	13	14			17														27	27	27
																					19	19									26	26			
					5									13	13					18														27	
																	15						20					24	25		26				
1	1											11	12	13						18	19	19													
1														13						18															
					5							11	12				15						20					24	25		26			27	
	1				5		6		8	9	10	11	12		13	14	15	16	14				20			22		24	25		26			27	
		2	3	4		6			8									16						21	22		23			26			27		
								7																											
														13						18															27
																					19	19										26			
1																																			

Stack Redesign

What did rethinking the entire computing stack at once get us?

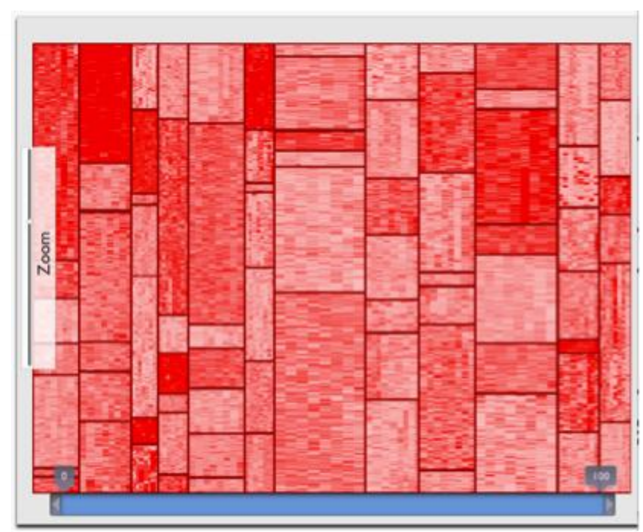
- Productivity programs can create applications that require efficiency
  - Scale
  - Performance
- Easily target many platforms and features
  - Example: Vector units on RISC-V Chip
- Efficient performance predictibility
  - Interactivity and Responsiveness
  - Realtime Performance

Integrated Demos

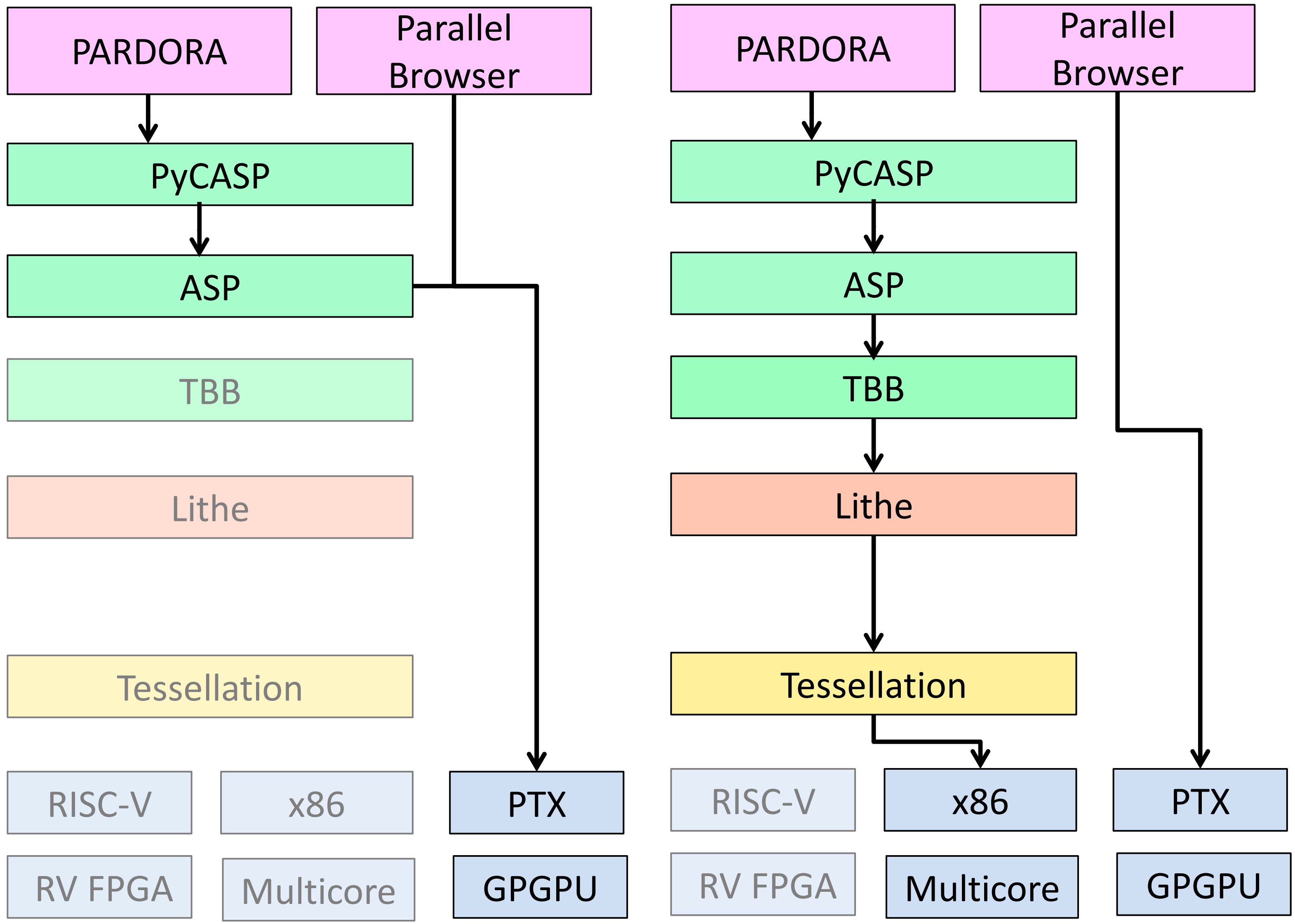
- Two Demos to show off the ParLab Stack
- Fun and compelling applications that require efficiency
- Easily target many platforms and features
- Interactivity and Realtime Performance
- Integration!

Music Exploration and Recommendation

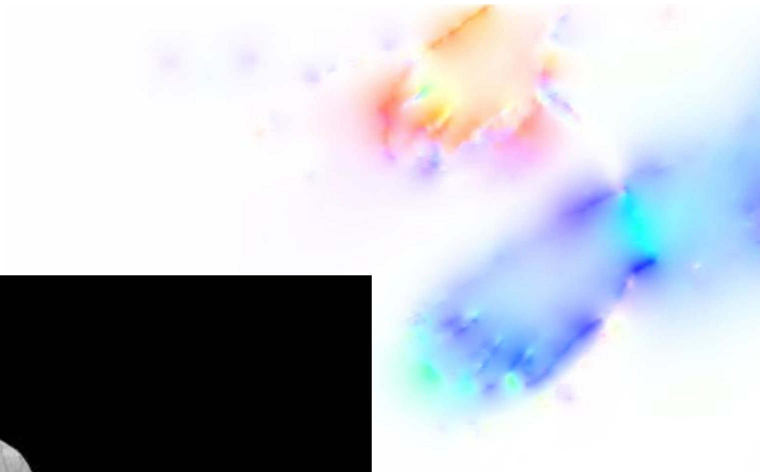
- Better Pandora
- Audio Content Analysis Framework
- Parallel Browser Big Data Visualization
- Demonstrates Scale and Responsiveness



Music Recommendation and Exploration

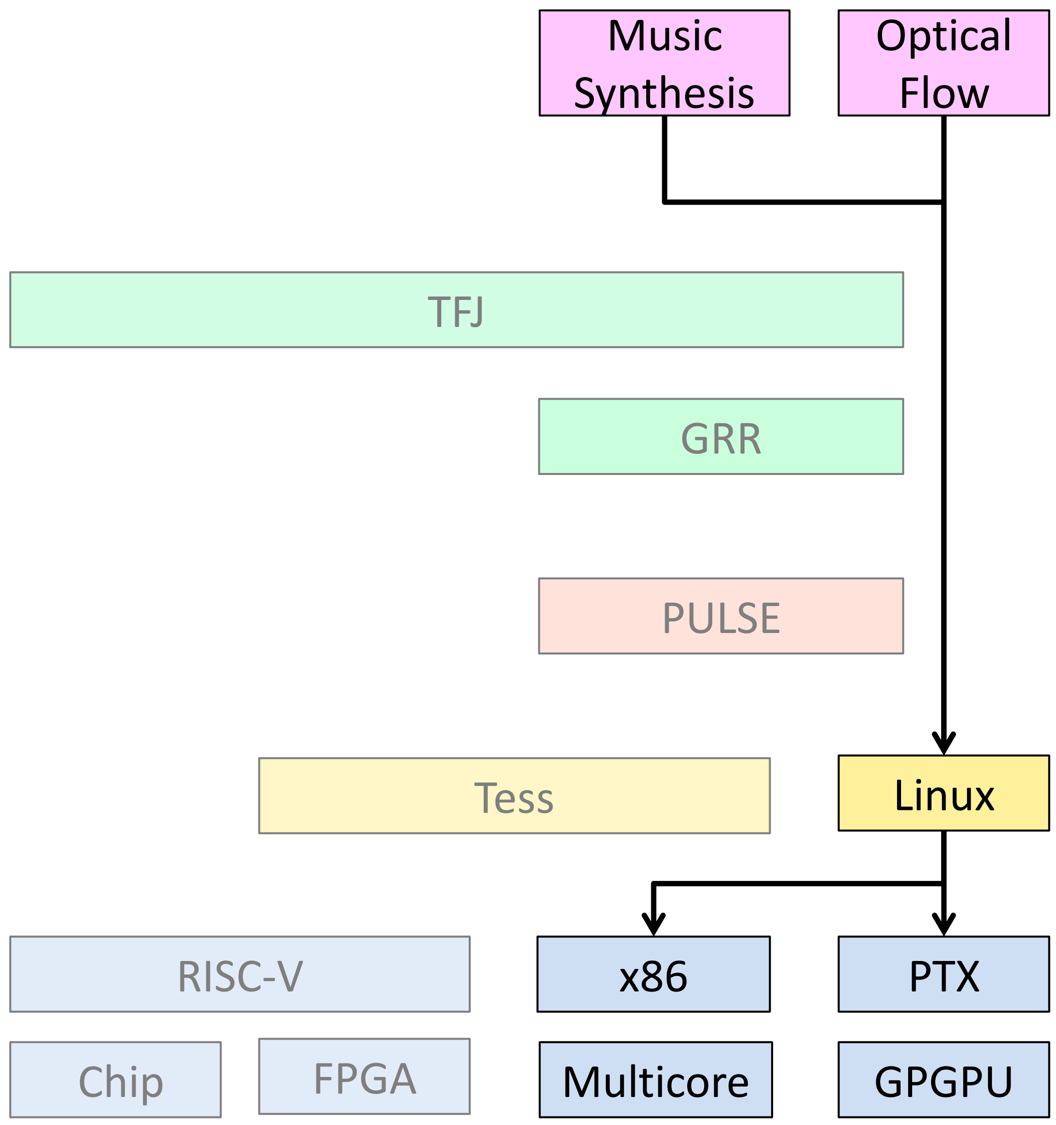


Virtual Musical Instrument

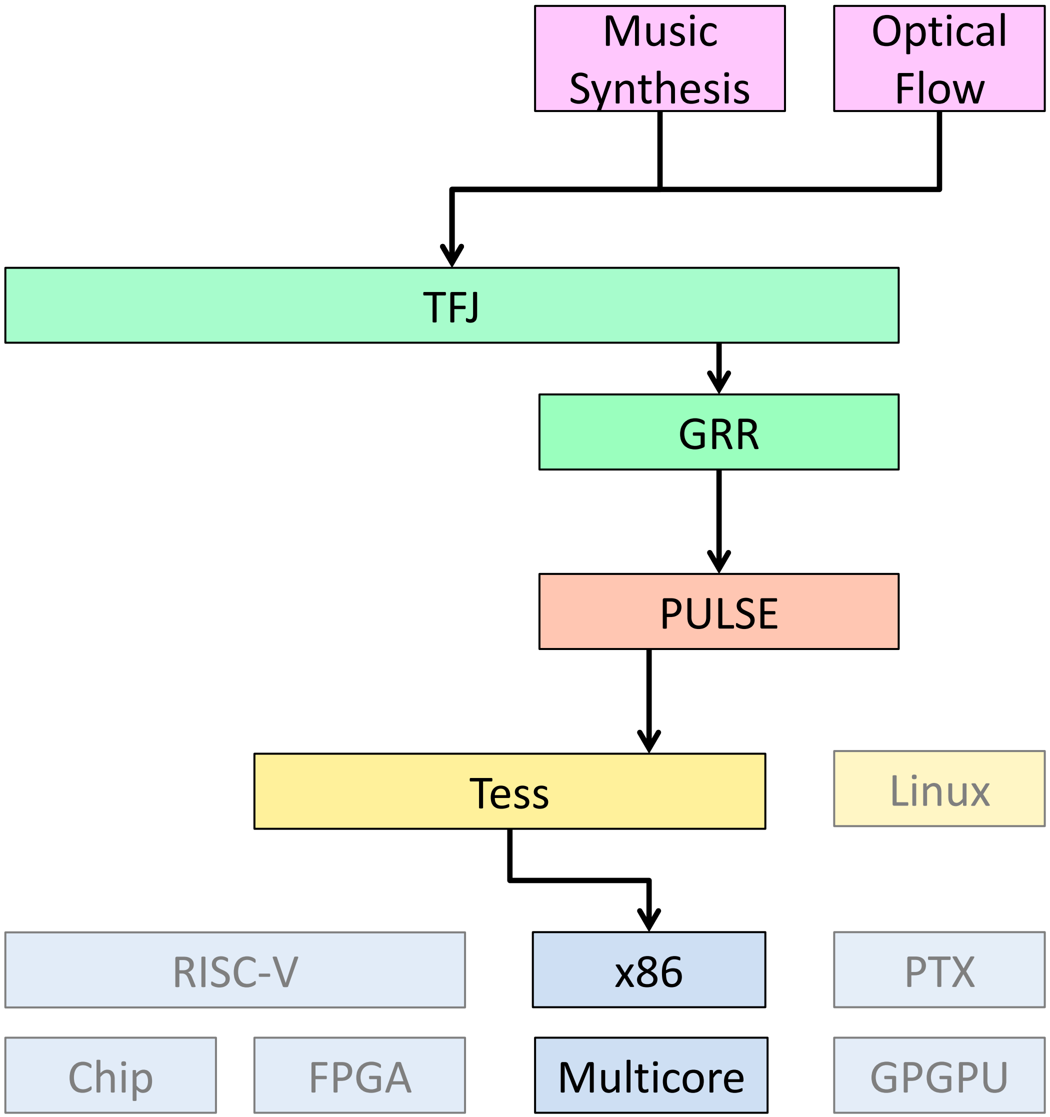


- Musical Instrument using a camera
- Music Synthesis
- Vision Applications
- Demonstrate Realtime

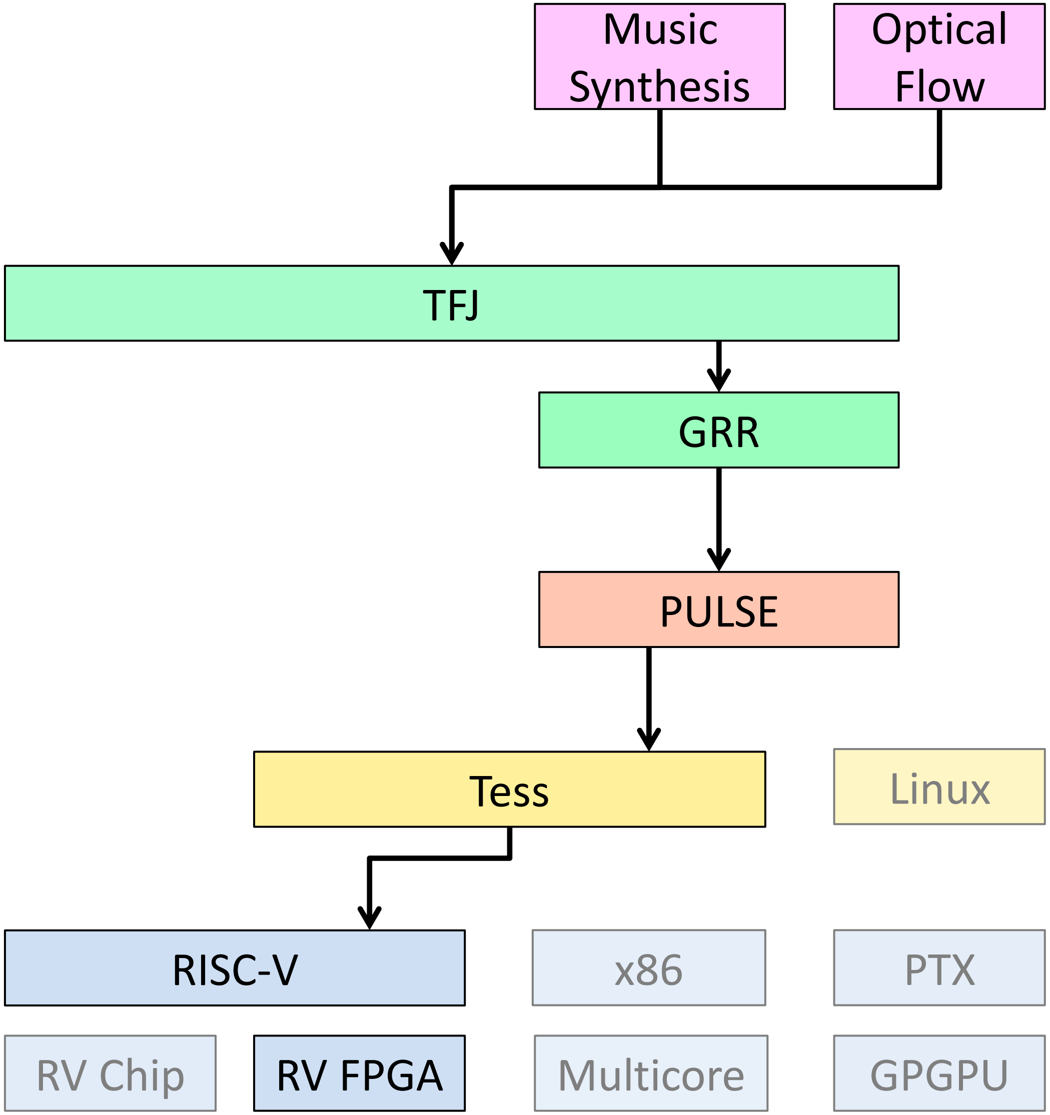
Virtual Musical Instrument



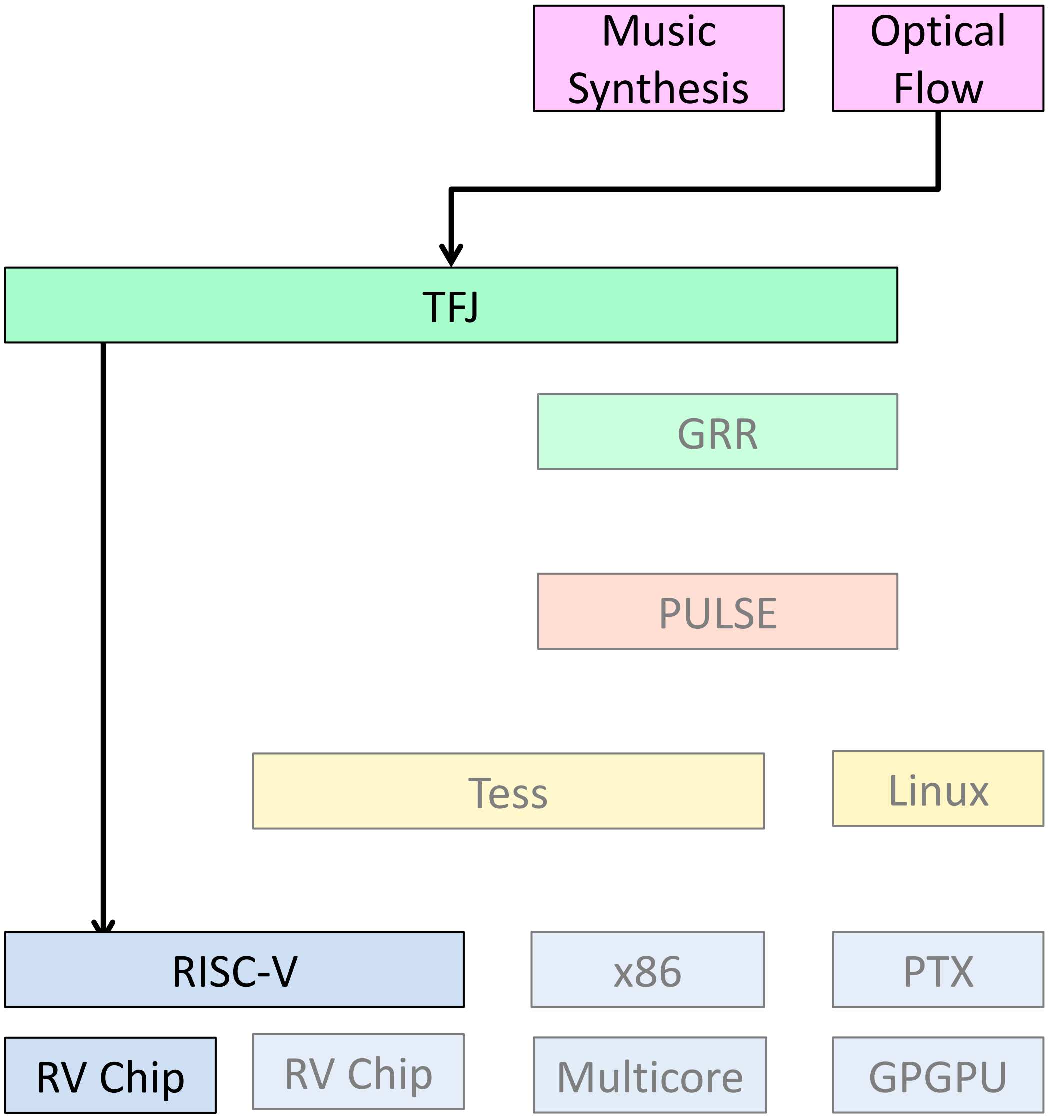
Virtual Musical Instrument



Virtual Musical Instrument



Virtual Musical Instrument



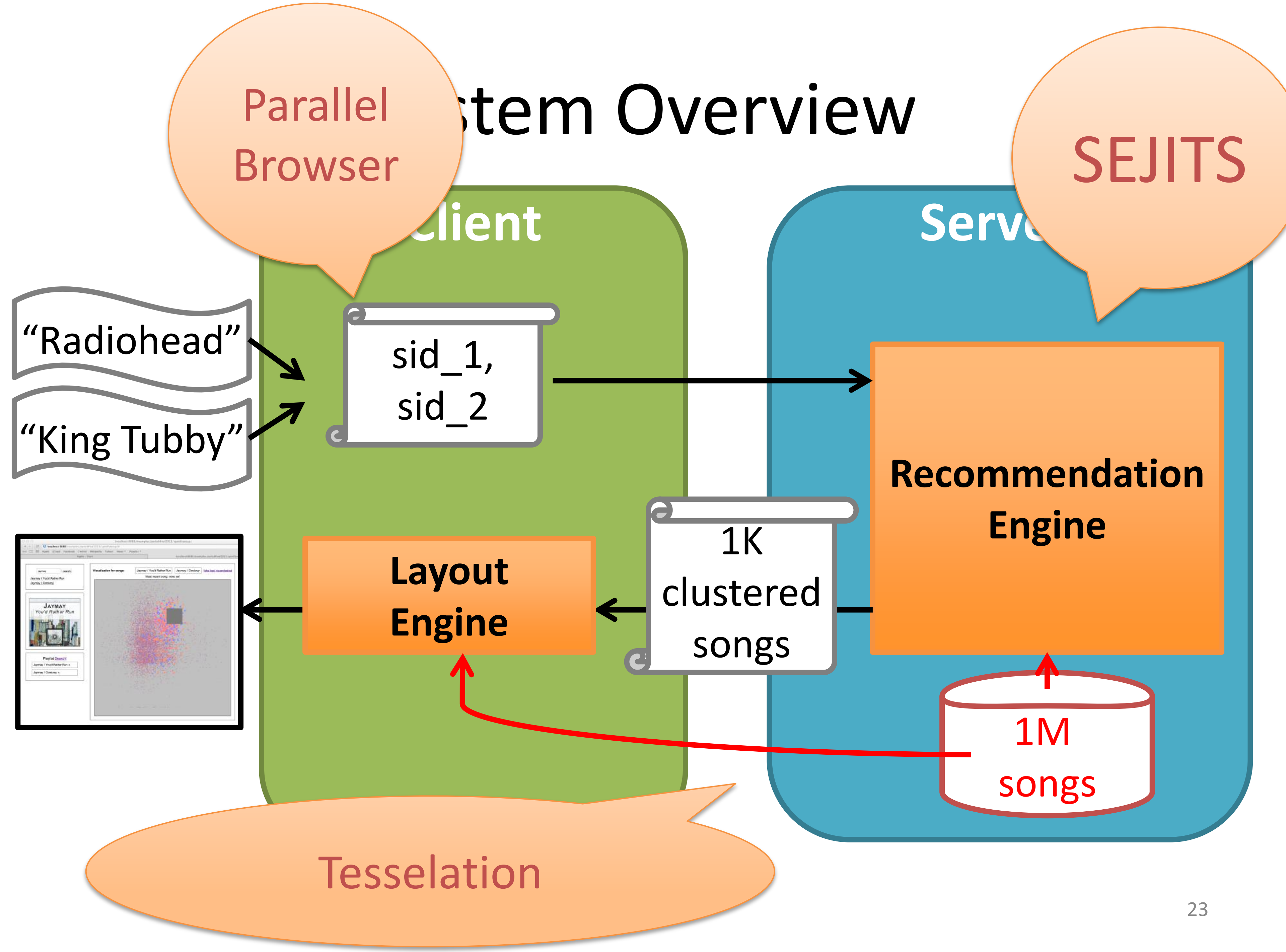
Music Recommendation & Exploration Demo

Leo Meyerovich, Katya Gonina, Gage Eads, Eric Roman, Eric Battenberg, Henry Cook, Gerald Friedland

End of ParLab Celebration

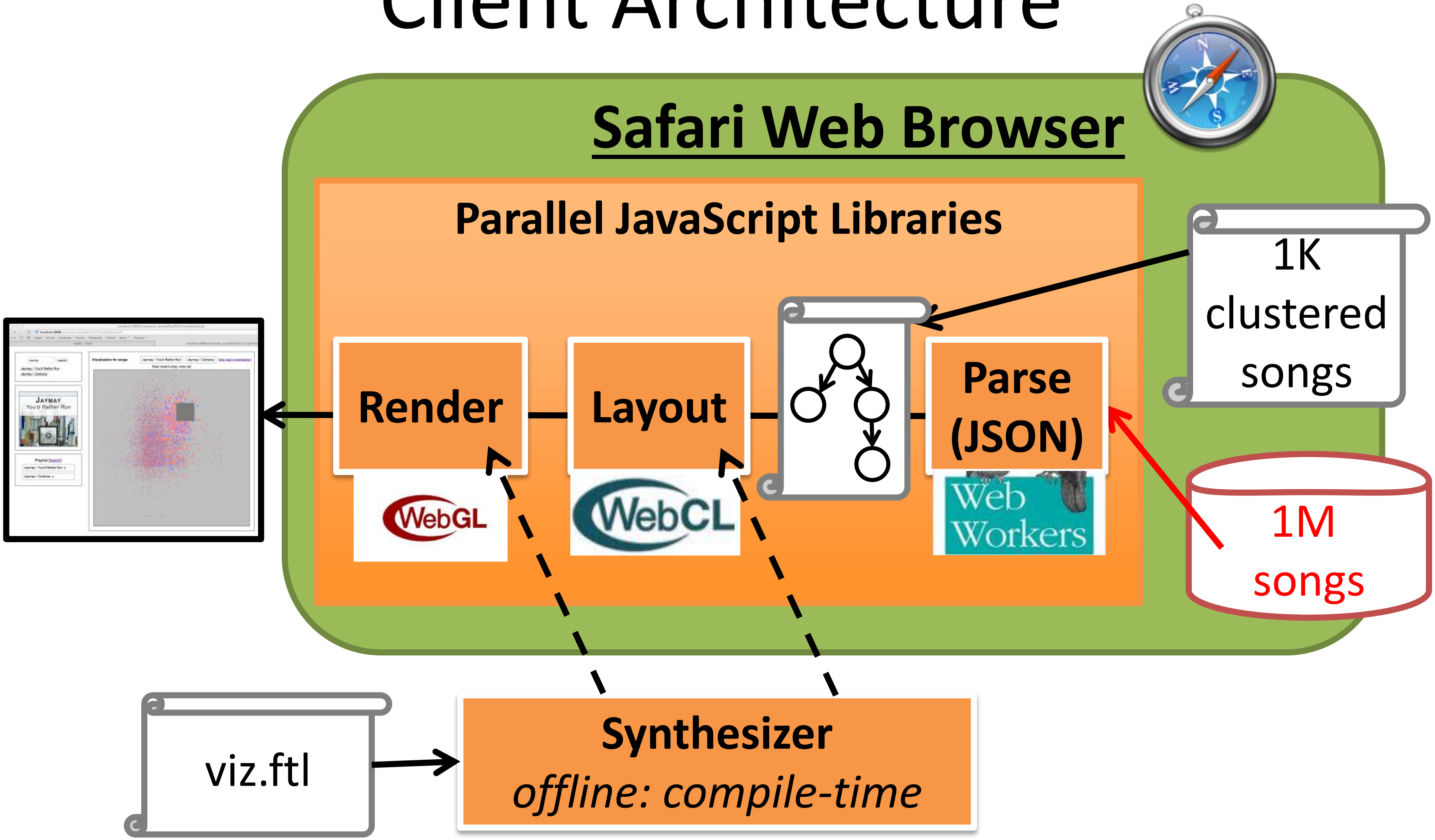
May 30, 2013

Demo

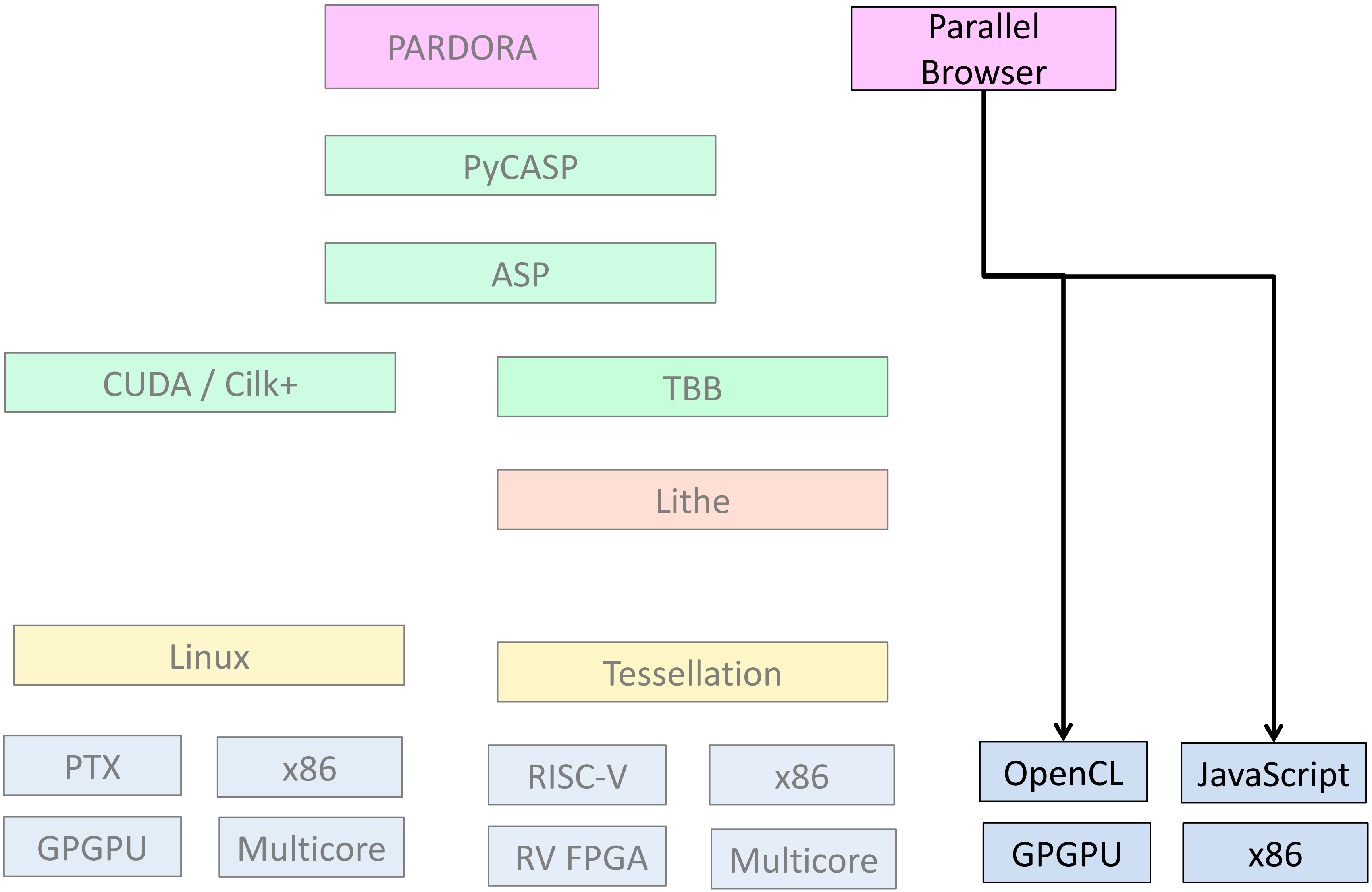


stem Overview
Parallel Browser
SEJITS
Client
Serve
“Radiohead”
“King Tubby”
sid_1, sid_2
Recommendation Engine
1K clustered songs
Layout Engine
1M songs
Tesselation

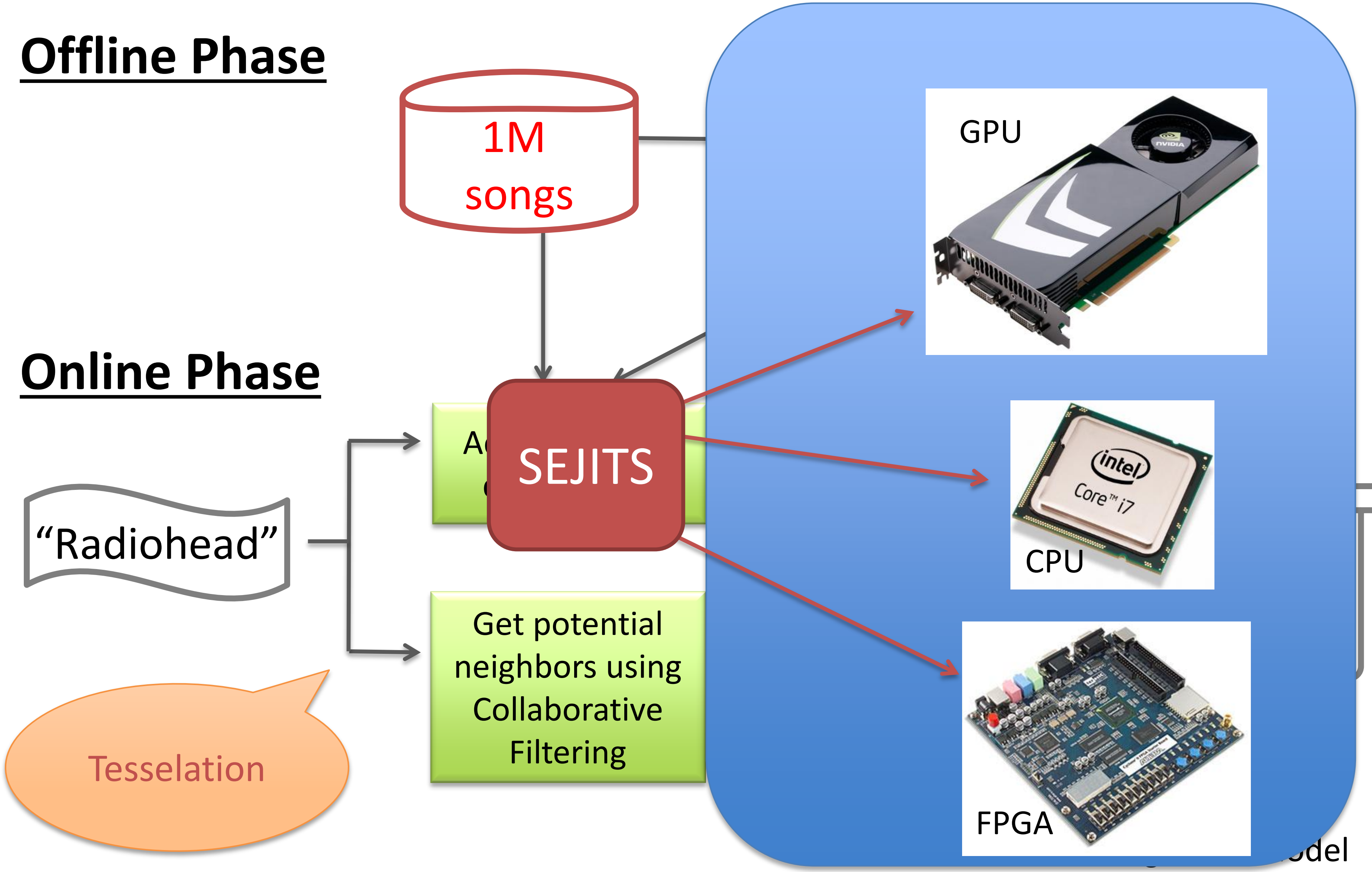
Client Architecture



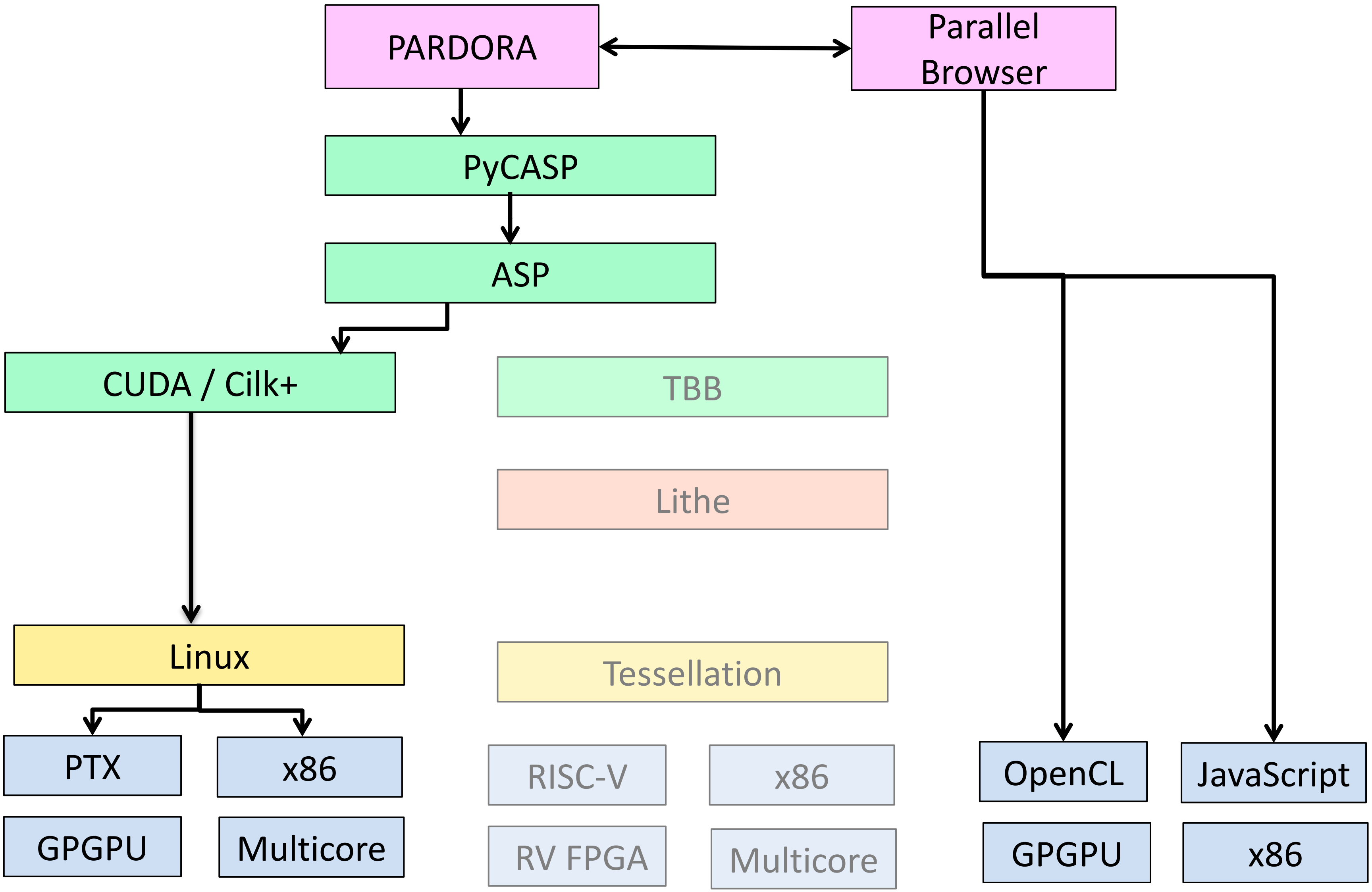
Music Recommendation Stack



Server Architecture



Music Recommendation Stack



Music Recommendation Stack