

New Resilient Architecture

Our new architecture is fault-tolerant. It enables Pepper robots to simultaneously run real-time dense RGB-D SLAM, Speech Recognition, Human Body and Face descriptions and Object Detection over ROS all on the onboard CPU.

**Human-robot Interaction:** human gesture and pose, object detection and recognition, face recognition and human features detection based on deep learning with automatic efficient local neural network fallback with custom in-context tablet interfaces. Generic and transparent implementation and execution directly on Pepper.

**Magic\_ROS** enables robust and efficient communication in faulty networks; automatic dispatch of work to GPU or local running servers with configurable strategies; flexible node data sharing using any native Python datatypes (no need for custom messages); and automatic node diagnostics, logging and introspection, user-friendly errors about the environment and network conditions.

**Navigation:** running real-time on board Pepper. Real Time Appearance Based Mapping using RGB-Depth (Mapping, Localisation); Timed Elastic Bands optimal trajectory planner (Planning, Obstacle Avoidance), move\_base\_flex (custom implementation).

**Speech Recognition:** our custom grammar searches over Google Speech with Sphinx local fallback using PocketSphinx.

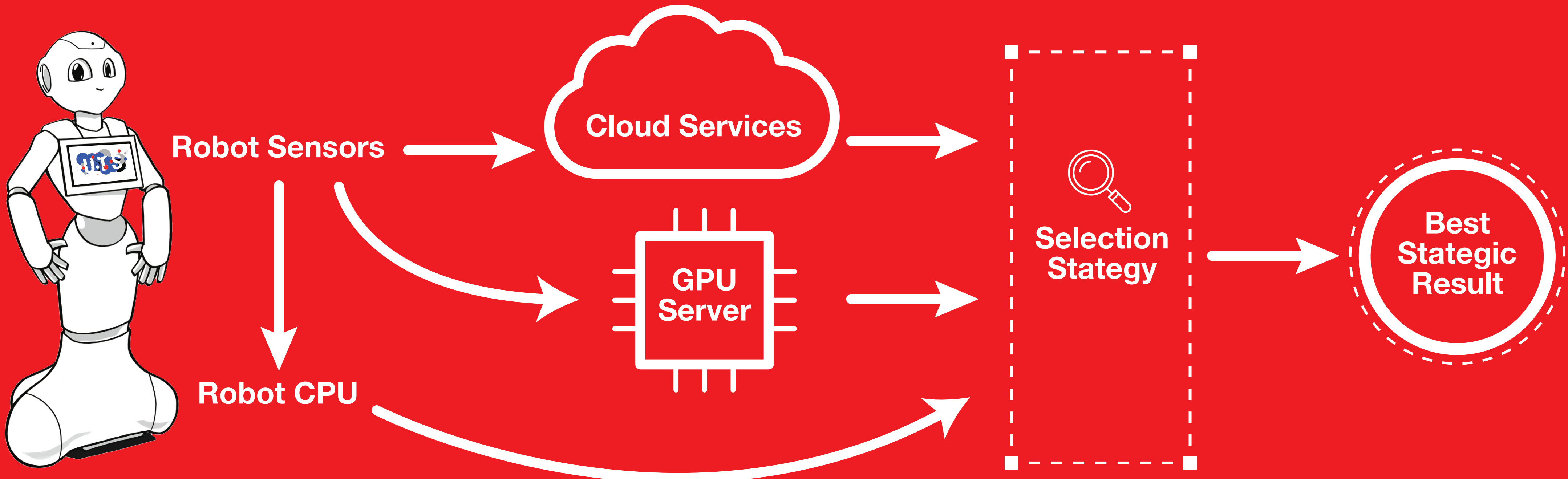
**Object Recognition:** dark\_magic uses Darknet on GPU server; Darknet fallback on Pepper.

**Person Recognition and Location:** magic\_poses uses OpenPose and dlib's face\_recognition for accurate and feature rich detections running on an external GPU server with local optimized fallback. Shirt segmentation with occlusions and colour detection. Find 3D location of people with OpenPose even with partial occlusions.

**Face Detection:** magic\_faces recognition of human facial features by re-using face encodings from facial recognition with custom trained SVMs.



Robustness is achieved using adaptive strategies to select the most reliable strategic result from the relevant local & remote computing nodes over unreliable networks.



Magic ROS: A New Framework for Robust, Reliable and Adaptive Social Robot Services.

Development Tools and Processes

We use reproducible methods to build our robot operating systems and software packages. Docker image for building packages; Local Database of Skills: General Pepper skills; RoboCup skills.

Userspace Operating System

magiclab\_pepper\_os

Allows the latest modern software to be run directly on Pepper using a Gentoo Prefix rolling release distribution built nightly via Docker on cloud services. With a simple folder copy-paste (does not require root access) frameworks, libraries and tools are ready to use promoting rapid robust development.

On-Site Automated Testing and Packaging

- GitLab source code repository
- GitLab-Runner build farm (matrix builds for multiple architectures and distros)
- Private Pip repository
- Private Debian repository

Analytics and Preflight Checklist

We use sensor, implementation and execution data collection for application analytics. Each application displays its health using standard ROS diagnostic messages, on a custom interface that runs on Pepper's tablet.

Diagnostics Viewer: Connected to ROS Bridge ws://192.168.1.11.9090/

generic_diagnostics_chocolate		
CPU: 58% (6.68, 5.56, 3.66)	OK	
► 9 values		
DISK: 6736 MB Available [Cached]	OK	
► 157 values		
Hostname: chocolate [Cached]	OK	
► 1 value		
Internet: 138.25.211.92 [Cached]	OK	
► 2 values		
NET: 198.18.0.1	OK	
► 11 values		
RAM: 46.8% Used   2105MB Available	OK	
► 11 values		

robocup_diagnostics		
Magic Listen: Not Running		Error
▼ /magic_listen_capture Running		
/magic_listen_pocketsphinx_transcribe Not Running		
Magic Speak: Running	OK	
► 1 value		
Object Recognition: Not Running	WARN	
► 1 value		
OpenPose: Running	OK	
► 1 value		

