

# Development of NEST Challenge Application: Distributed Pursuit Evasion Games (DPEGs)

The  $S^5$  Group

**S. Sastry, S. Schaffert, L. Schenato, S. Simic, B. Sinopoli**

**with E. Brewer, D. Culler, D. Wagner**

**University of California, Berkeley**

**{sastry,sms,lusche,sinopoli}@eecs.berkeley.edu**

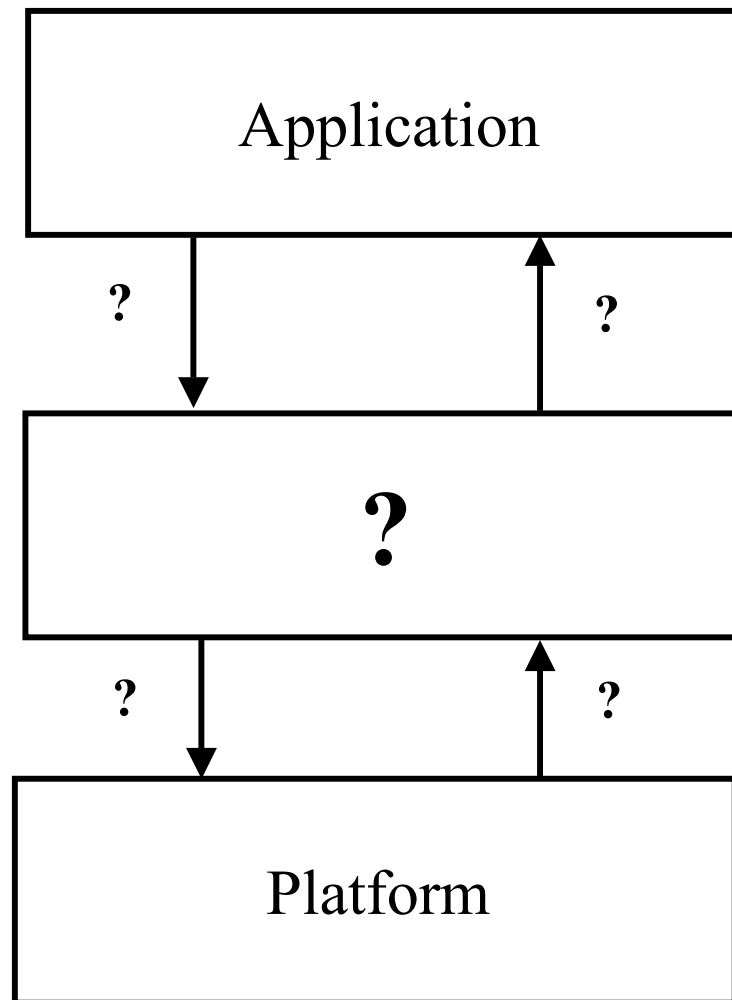
# Outline

---

- **Approach to design**
- **Current PEG setup**
- **Sensor network enhanced PEG**
- **Platform**
- **Application definition & challenges**
- **Component based architecture**
- **Basic parameters and performance metrics**
- **Implementation steps**
- **Other challenge applications**
- **Appendix: Platform specifications**

# Approach to design

---



# Open Experimental Platform

---

- **Open:** software architecture has been made available to the community
- **Experimental:** we cannot expect that algorithms will work at the very first time. Trial and error approach is the winning approach in helical development
- **Platform:** the motes are now available to all teams. Sensor platform can be easily replicated

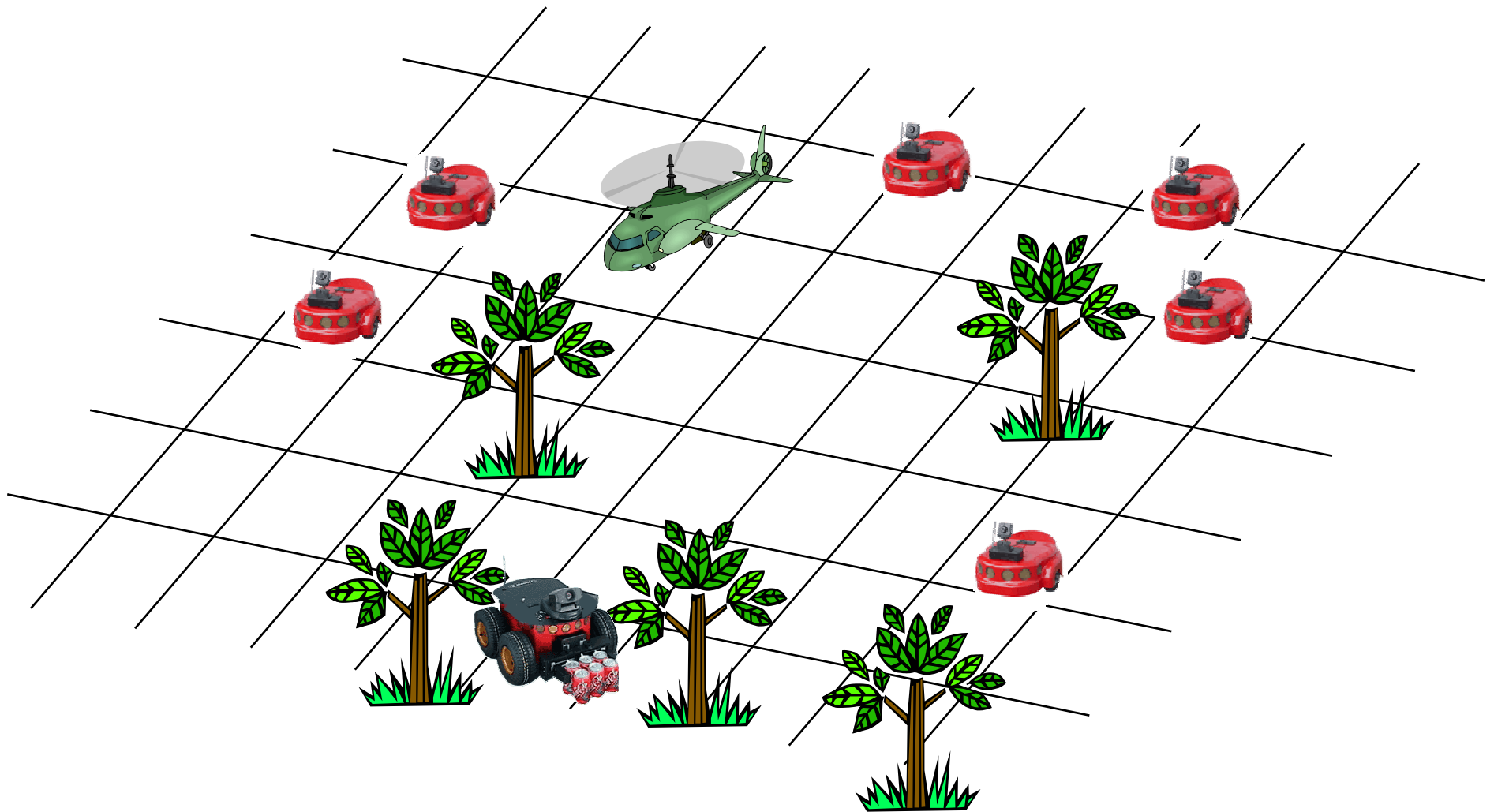
# Requirements

---

- **Top Down approach:**
  - Application will provide specifications to network services and components
- **Bottom Up approach:**
  - Platform limitations will define the high level specifications
- **Our Approach encompasses the two views:**
  - We shall provide specifications
  - Platform limitations will provide feedback in order to tune these specifications

# The State-of-the-Art

---

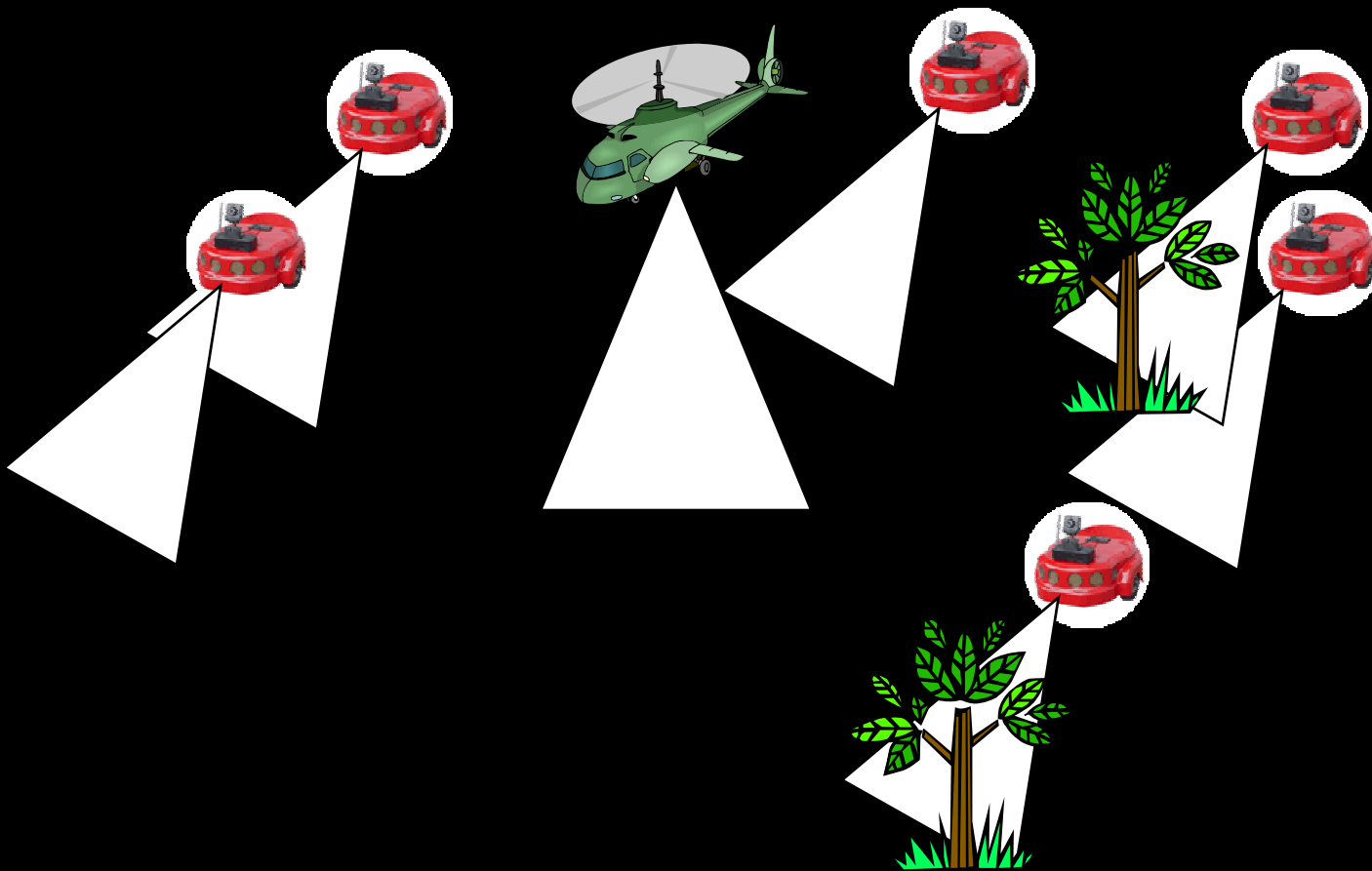


February 12, 2002

Nest PI Meeting

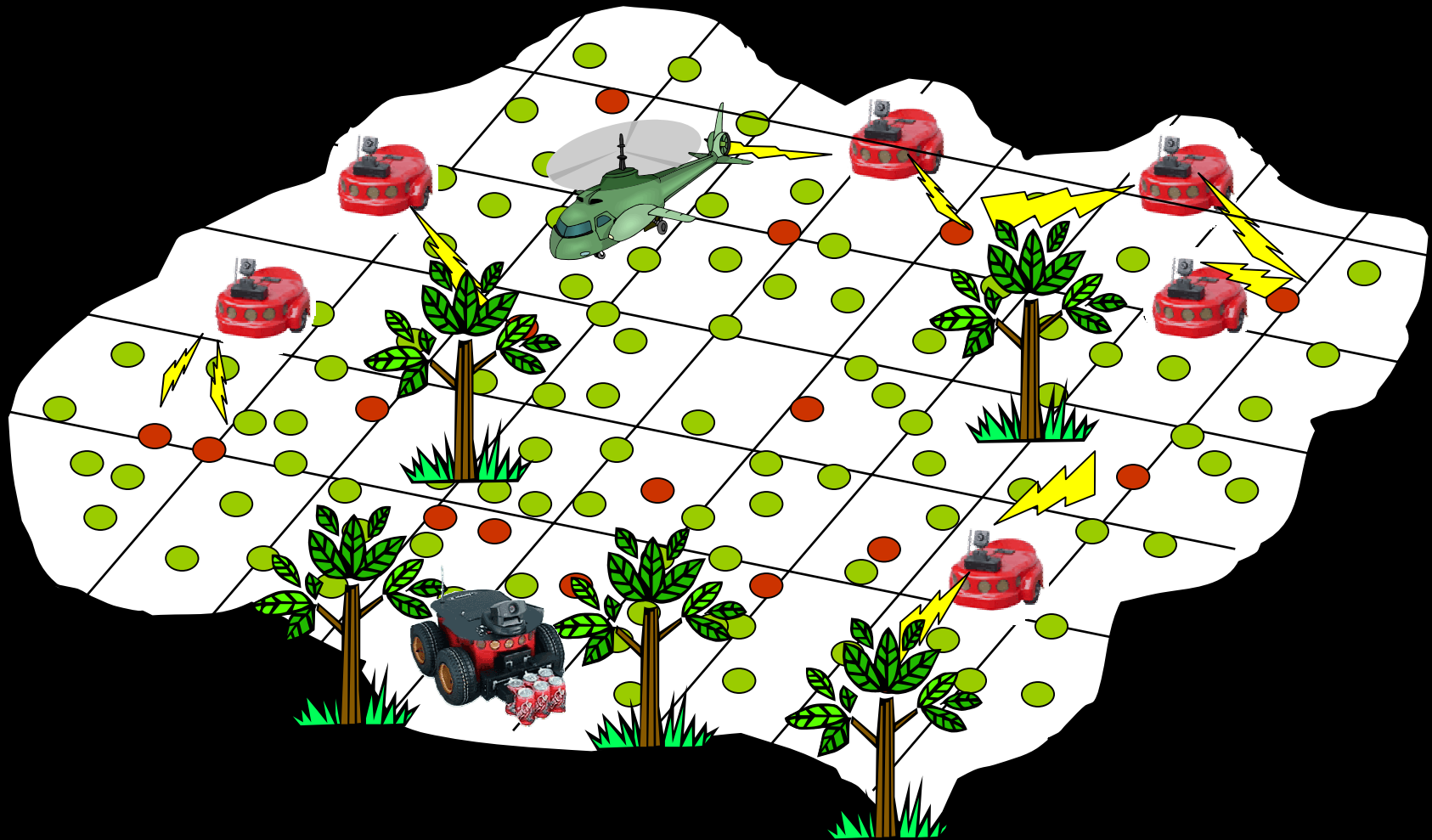
# What pursuers really see

---



# Sensor net increases visibility

---



February 12, 2002

Nest PI Meeting

8

# What a sensor network can do for PEG

---

- **Potential Issues in current PEG**
  - Cameras have small range
  - GPS jamming, unbounded error of INS, noisy ultrasonic sensors
  - Communication among pursuers may be difficult over a large area
  - Unmanned vehicles are expensive
    - » It is unrealistic to employ many number of unmanned vehicles to cover a large region to be monitored.
  - A smart evader is difficult to catch
- **Benefits from sensor network**
  - Large sensing coverage
  - Location aware sensor network provide pursuers with additional position information
  - Network can relay information among pursuers
  - Sensor network is cheap and can reduce number of pursuers without compromising capture time
  - Sensibly reduce exploration of the environment
  - A wide, distributed network is more difficult to compromise

**Overall Performance can be dramatically increased by lowering capture time, by increasing fault tolerance and making the pursuer team resilient to security attacks**

# Outline

---

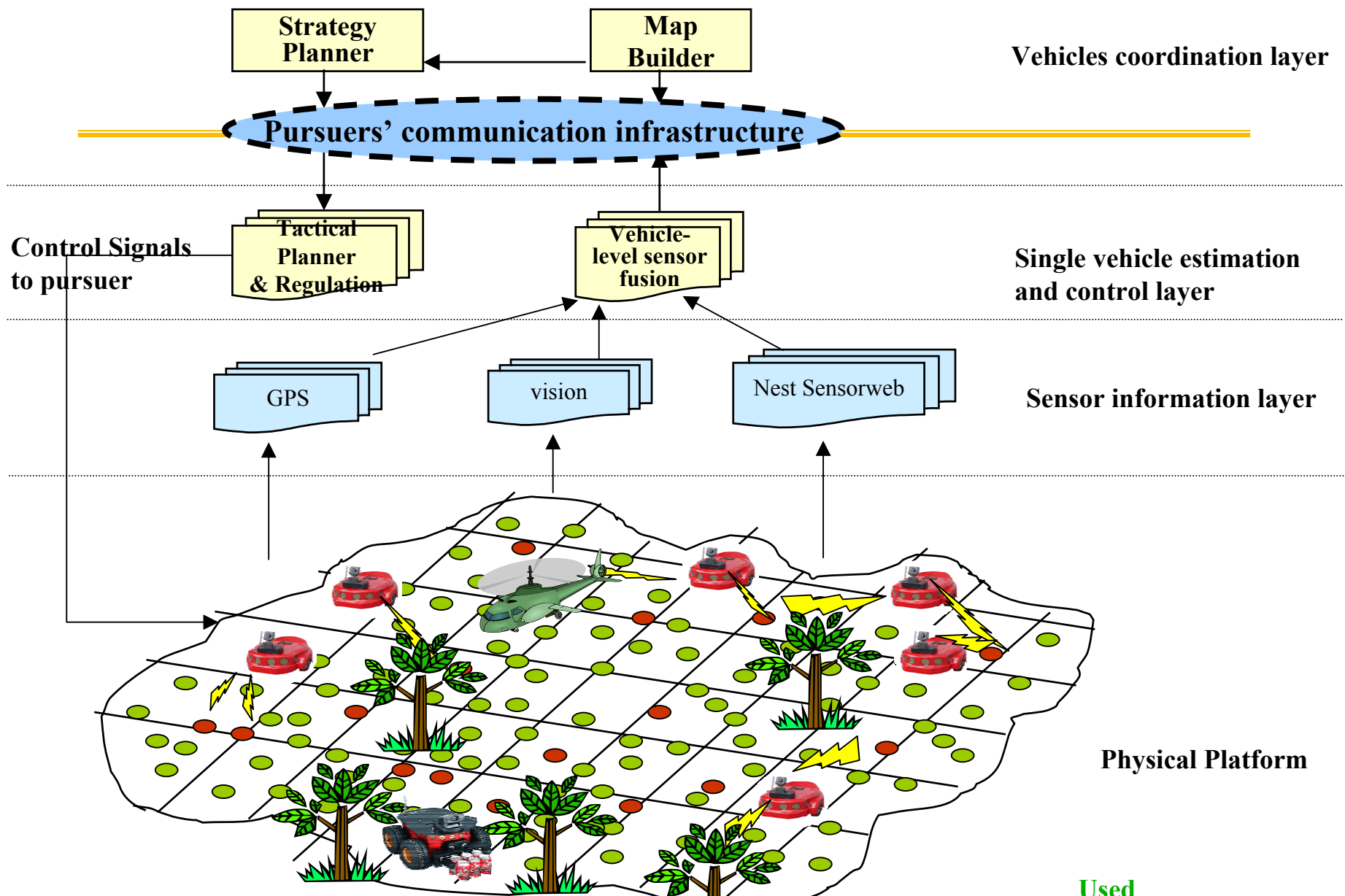
- Approach to design
- Current PEG setup
- Sensor network enhanced PEG
- **Testbed definition and challenges**
- Component based architecture
- Basic parameters and performance metrics
- Implementation steps
- Other challenge applications
- Appendix: Platform specifications

# Definition of the testbed\*

---

- **A level field (400-2500 m<sup>2</sup>) with 5-15 tree-like obstacles**
- **Pursuers' team**
  - 400-1000 fixed wireless randomly placed sensor nodes with at least two modes of sensing (acoustic, magnetic)
  - 3-5 ground pursuers,
    - » Equipped with GPS, onward looking camera, ultrasonic sensors, wavelan based communication infrastructure
  - 1-2 aerial pursuers moving through the field
    - » GPS, INS, downward looking camera, wavelan communication capabilities.
- **Evaders' team**
  - 1-3 ground evaders, with same equipment as ground pursuers

\*See appendix for robots specifications



# Challenge problems to be addressed

---

- **Self organization of motes into a sensorweb**
  - Creation of a communication infrastructure
  - Self-localization
  - Synchronization
- **Tracking of evaders' by pursuers' team**
  - Evaders' position and velocity estimation by sensor network
  - Communication of sensors' estimates to ground pursuers
- **Design of a pursuit strategy**
  - Coordination of ground & aerial pursuers
- **Network maintenance**
- **Robustness**
- **Security**

# Closed-loop at many levels

---

- **Within a node**
  - Algorithms adapt to available energy, physical measurements, network condition
- **Across the network**
  - discovery and routing, transmission protocols are energy aware and depend on application requirements
- **Within the middleware components**
  - synchronization, scheduling, localization
- **On the vehicle**
  - direction, stability, probabilistic map building
- **Among the vehicles**
  - competitive, hidden Markov decision processes

Used

# Coordinated pursuit strategy

---

- **Estimation of number of evaders**
  - Disambiguation of multiple signal traces
- **Estimation of capture time: several possibilities**
  - Every pursuer gets the closest evader
  - Pursuers relay partial info about evaders to base station
  - Base station estimate time-to-capture and assign evaders to pursuers
  - Pursuers communicate with each other locally and implement a distributed pursuit strategy
- **Vision-based tracking**
  - Pursuer switch to vision-based tracking when evader is within camera range

# Pursuit Evasion Games specifications

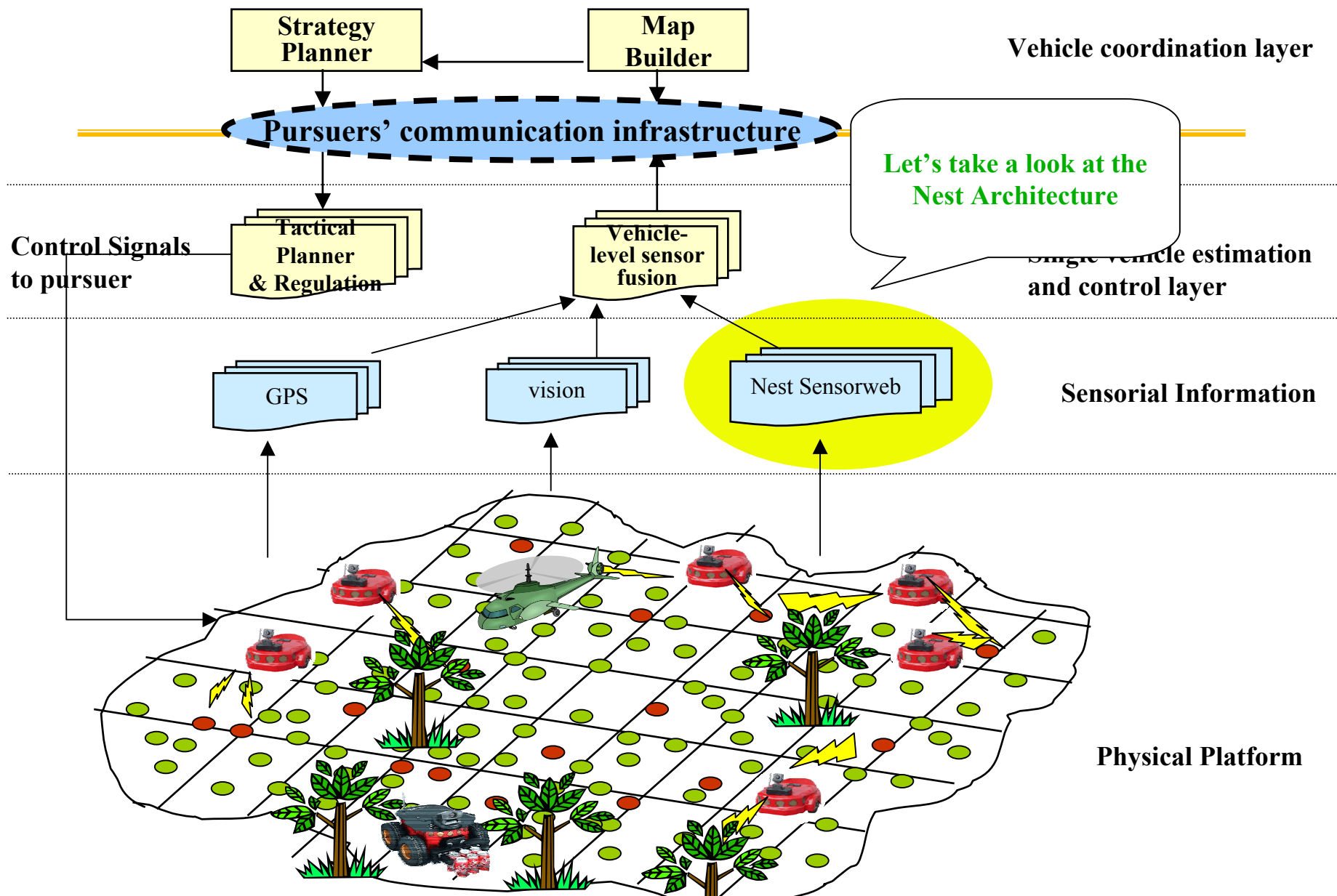
---

- **The goal is to minimize the time necessary to catch the evaders, i.e. having a ground pursuer within a certain distance from an evader**
- **Other possible performance metrics to optimize for (minimize) are:**
  - **Total energy spent**
  - **Given a number of evaders, minimize number of pursuers needed with respect to a constant average time to capture**
  - **Degradation of performance (average capture time) in view of:**
    - » **Percentage of corrupted nodes**
    - » **Percentage of failing nodes**
    - » **Smart evaders**

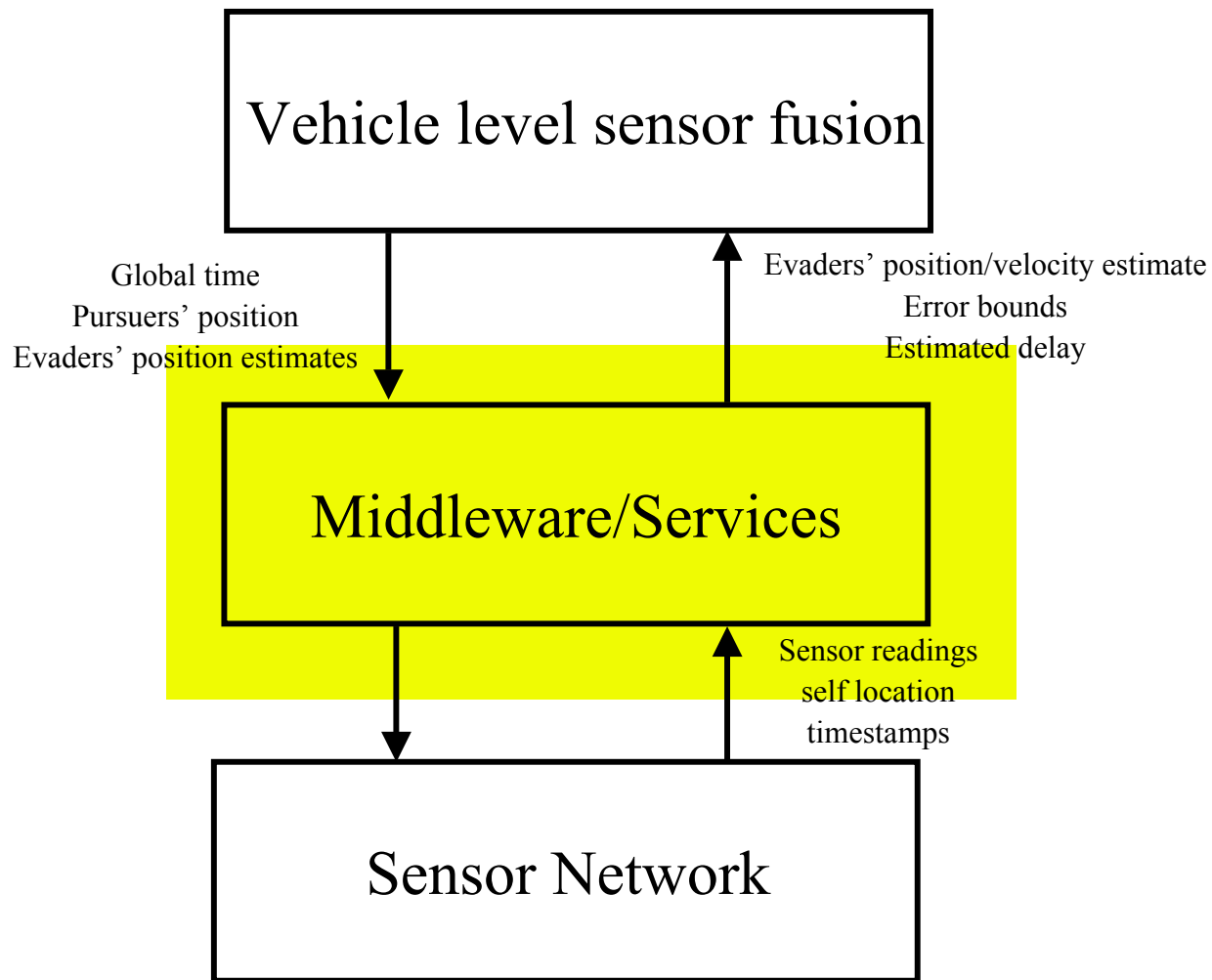
# Outline

---

- Approach to design
- Current PEG setup
- Sensor network enhanced PEG
- Testbed definition and challenges
- **Component based architecture**
- Basic parameters and performance metrics
- Implementation steps
- Other challenge applications
- Appendix: Platform specifications

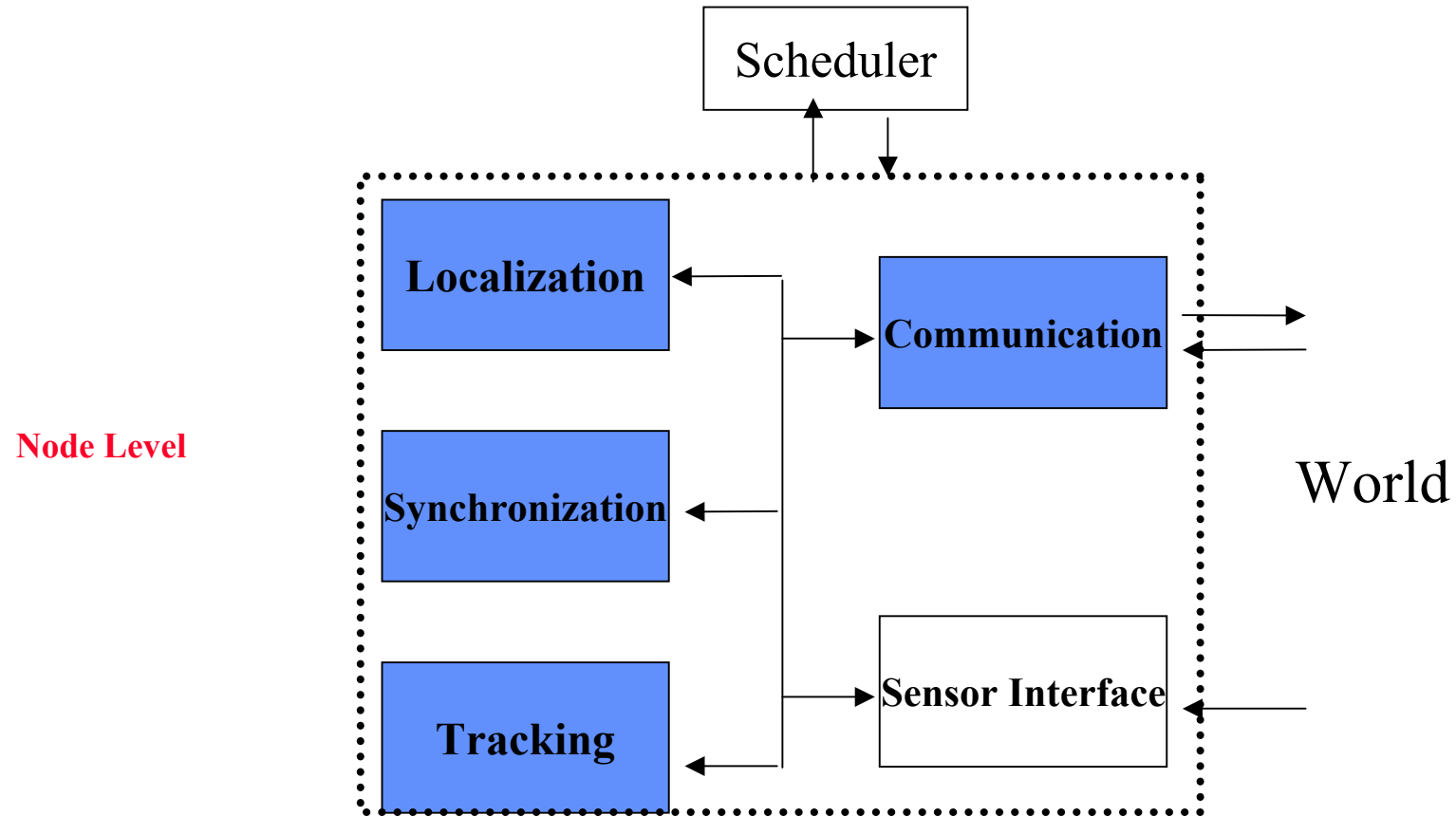


# Nest architecture: modular modeling



# Middleware Component Architecture

---



# Time bounded synthesis (vehicle level)

---

- **Reconfiguration at runtime caused by:**
  - Parameters of the problem change in time
  - Degradation of mote parameters:
    - » Density decreases due to motes failure
    - » Communication range is lower when battery is low
  - Optimal function or constraints vary
  - Estimation performance degrades due to compromised motes
  - System cannot satisfy the constraint imposed by application
  - New control schemes are to be implemented in a changing environment

# Coordination services and real-time synthesis (summary)

---

- **Coordination algorithms**
  - Localization
  - Time synchronization
  - Tracking
  - Data consistency check to spot compromised, malfunctioning nodes
- **Real-time synthesis**
  - Real time scheduling of services to be performed
  - Real time reconfiguration due to:
    - » Compromised or faulty nodes
    - » Reprioritization of optimality metrics

# Outline

---

- Approach to design
- Current PEG setup
- Sensor network enhanced PEG
- Testbed definition and challenges
- Component based architecture
- **Basic parameters and performance metrics**
- Implementation steps
- Other challenge applications
- Appendix: Platform specifications

# Tracking

---

## Requirements for PEG

- Position/Velocity Estimate Update Frequency: 1-10Hz
- Average Delay Time\*: 0.1-0.5s
- Position Accuracy Radius: 0.05-0.5m
- Velocity Accuracy Radius: 0.05-0.10 m/s

## Parameters

- Average nodes density:  $\rho$
- Minimum Sampling period:  $T$
- **Evader** position estimation error variance:  $\sigma_e$ 
  - Estimation delay:  $\mathcal{T}$
- **Evader**
  - Maximum speed:  $v_e$
  - Maximum acceleration:  $a_e$
- Pursuer maximum speed:  $v_p$

## Other metrics to optimize (minimize) for

- Number of packets sent per node
- Energy expenditure per node
- Degradation of accuracy vs. percentage of corrupted or dead nodes
- Node density vs. accuracy

# Localization

---

## Requirements for PEG

- Position Accuracy Radius of each Node: 0.02-0.05m

## Other metrics to optimize (minimize) for

- Number of packets sent per node
- Convergence time
- Position accuracy vs. nodes rearranging
- Energy expenditure per node
- Degradation of accuracy vs. percentage of corrupted or dead nodes
- Node density vs. accuracy

# Communication

---

## Requirements for PEG

- **Maximum tolerable delay in communication between any two nodes in the network : 50 ms**
- **Maximum tolerable delay between any node and any ground pursuer: 400 ms**

## Other metrics to optimize (minimize) for

- Network stability vs. nodes rearranging (need a quantitative measure)
- Energy expenditure per bit transmitted
- Loss Probability

# Synchronization

---

## Requirements

- **95<sup>th</sup> percentile clock offset: 0.005 seconds**

## Parameters

- **Percentage of nodes with absolute clock (including pursuers)**

## Performance Metrics

- Number of packets sent per node
- Convergence time
- Accuracy vs. percentage of nodes with absolute clock (GPS nodes)
- Energy expenditure per node

# Outline

---

- Approach to design
- Current PEG setup
- Sensor network enhanced PEG
- Testbed definition and challenges
- Component based architecture
- Basic parameters and performance metrics
- **Implementation steps**
- Other challenge applications
- Appendix: Platform specifications

# Two stages implementation

---

- **First step:**
  - Use setup described in OEP exercise to test localization, communication, synchronization and tracking of a moving target
  - Nodes will be put on grid in the first stage
  - Nodes will be randomly distributed in a second stage
  - Test fault tolerance, security of sensor network
- **Second step:**
  - Implementation of the full scale pursuit evasion game, as described above, in Richmond Field Station (RFS)
- **Main Differences:**
  - Outdoor vs indoor environment
  - Scale
  - High level assets are moving
  - One more degree of closed loop control is added by playing the game rather than monitoring exclusively

## **This implementations open several interesting control problems at many levels**

- **Field of nodes collaborate with video system to perform ranging and localization to create coordinate system**
- **Build routing structures between field nodes and cameras**
- **Selection of low-level assets per object over time**
  - determined by local sensor processing and high-level coordination
- **Selection of high-level assets over time**
  - determined by in-coming data and higher processing
  - determines dynamic routing structure over time
- **Targeting of high-level assets**
- **Sensors guide video assets in real time**
- **Video assets refine sensor-based estimate**
- **Network resources focused on region of importance**
- **Control of coordination between pursuers to catch evaders**

# Timeline

---

- **First step implementation:**
  - **June '02:**
    - » **Development of basic components to be used in tracking application, i.e. localization, synchronization, tracking, communication infrastructure**
    - » **Development of a network monitoring tool**
  - **August '02:**
    - » **Testing of tracking application on a uniform grid of 50 nodes**
    - » **Test resilience or security properties of sensors net**
  - **January '03:**
    - » **Closed loop tracking demonstration on a randomly distributed sensor network.**
    - » **Coordination between sensor net and fixed cameras (high level assets)**

# Timeline continued...

---

- **Second step:**
  - **August '03**
    - » **Testing of pursuit evasion application outdoor on a 100-200 nodes sensor network with 2 ground pursuers, 1 aerial pursuer and 1 evader**
    - » **Cameras are not fixed anymore but mounted onboard the robots**
    - » **Test coordination among pursuers and sensor net**
  - **January '04**
    - » **Bigger net up to 400 nodes**
    - » **Use more than 1 evader allows to demonstrate efficiency of pursuers' coordination scheme**
  - **June '04**
    - » **Full scale Pursuit Evasion game (400-1000 nodes)**
  - **October '04**
    - » **Complete demonstration of goals achieved during the project**

# Outline

---

- Approach to design
- Current PEG setup
- Sensor network enhanced PEG
- Testbed definition and challenges
- Component based architecture
- Basic parameters and performance metrics
- Implementation steps
- **Other challenge applications**
- Appendix: Platform specifications

# Other Challenge Problems

---

- **Sensing and Updates of the Environment in response to Events and Queries.**
  - monitor the environment of a building and use this to instigate control actions such as lighting, HVAC, air-conditioning, alarms, locks, isolation, etc.
  - monitor and protect space from environmental attack
- **Distributed Map Building**
  - classic “art gallery” problem is provably hard
  - many agents with simple proximity sensors to detect obstacles
  - exchange info => dense collaborative map building

Tracking and Evader map building are two particular types of environmental monitoring and distributed map building respectively. Other groups are encouraged to define different application scenarios where environmental monitoring and map building can be experimented

# Outline

---

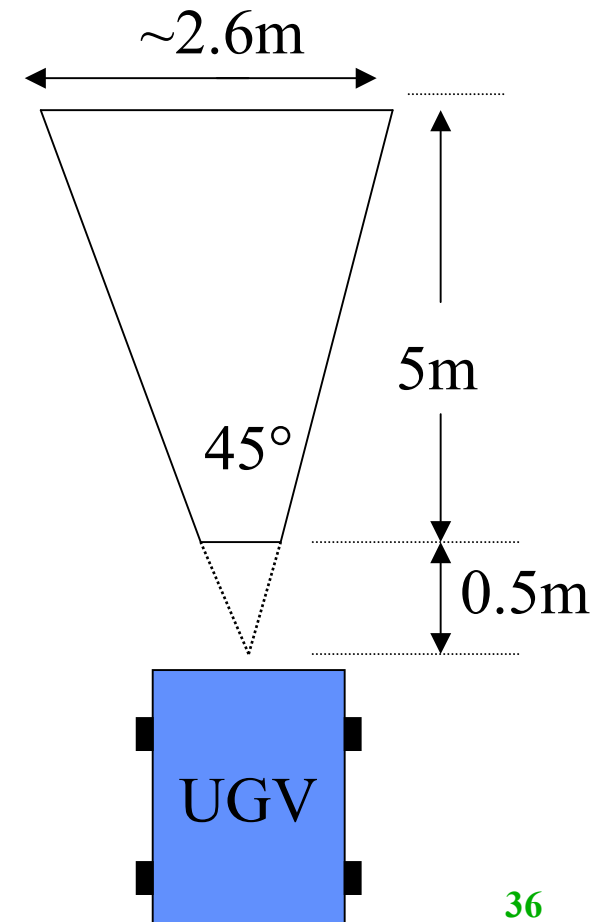
- Approach to design
- Current PEG setup
- Sensor network enhanced PEG
- Testbed definition and challenges
- Component based architecture
- Basic parameters and performance metrics
- Implementation steps
- Other challenge applications
- **Appendix: Testbed specifications**

# Ground Pursuer/Evader :Pioneer UGVs



- **Driving Speed: 0-0.5m/s**
- **Turning Speed: 90°/3sec**
- **Driving Time: 1 hour**
- **WaveLAN (IEEE 802.11b) RF Communication**
  - Frequency: 2.4 GHz
  - Range: 1 mile
- **Camera**
  - Pan  $\pm 90^\circ$
  - Tilt:  $\pm 30^\circ$
  - View Angle:  $45^\circ$
  - Zoom: 0-12x
- **GPS (Optional)**
  - Accuracy: 2cm radius

Camera Coverage (top view)

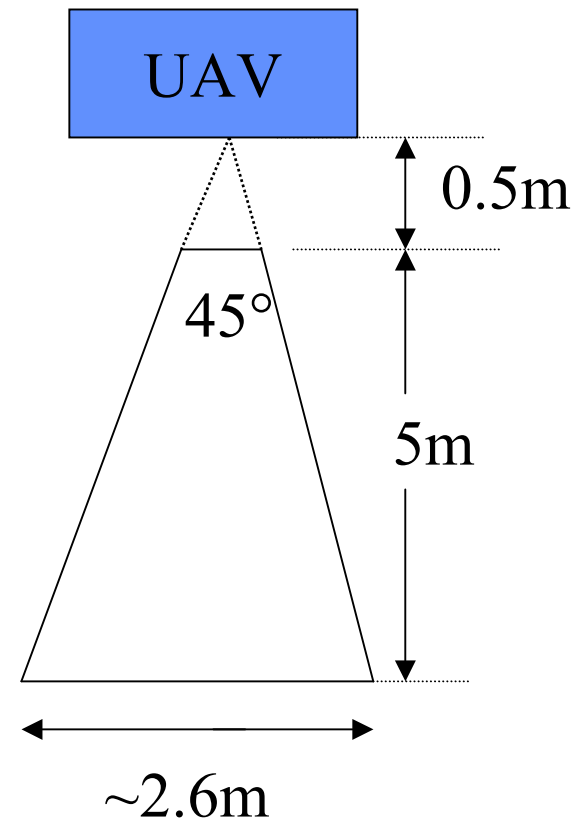


# Aerial Pursuer: Yamaha R50/Rmax UAV



- **Flight Speed: 0-5m/s (usually 1-2m/s)**
- **Turning Speed: 90°/5sec**
- **Flight Time: 1 hour**
- **WaveLAN (IEEE 802.11b) RF Communication**
  - Frequency: 2.4 GHz
  - Range: 1 mile
- **Camera**
  - Pan  $\pm 90^\circ$
  - Tilt:  $\pm 30^\circ$
  - View Angle:  $45^\circ$
  - Zoom: 0-12x
- **GPS (Optional)**
  - Accuracy: 2cm radius

Camera Coverage (side view)



## Field Nodes (“motes”)

---



- **Atmel ATMEGA103**
  - 4 Mhz 8-bit CPU
  - 128KB Instruction Memory
  - 4KB RAM
- **4 Mbit flash (AT45DB041B)**
  - SPI interface, 1-4 uJ/bit r/w
- **RFM TR1000 radio**
  - 50 kb/s
  - Sense and control of signal strength
- **Network programmable in place**
- **Multihop routing, multicast**
- **Sub-microsecond RF node-to-node synchronization**
- **Provides unique serial ID's**
- **On-board DC booster**
- **Sensor board: acoustic and magnetic sensors**

# Q&A

---

**<http://webs.cs.berkeley.edu>**