



# PSYCLONE 2.0

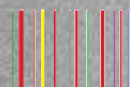
---

Beta Release 3



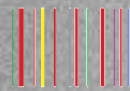
**COMMUNICATIVE  
MACHINES**

Powering the AI Revolution™



# What is Psyclone 2.0 ?

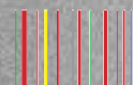
- Distributed platform for running and managing large modular systems
- Based on 15+ years of academic and commercial product development
- Mediated publish-subscribe server with direct module-to-module interaction
- Working with heterogeneous software modules communicating heterogeneous data
- Semantic message- and stream-based data exchange model
- Scalable as applications grow — from a single computer to large server farms



# Why Psyclone 2.0 ?

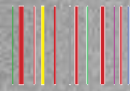
- Build & run industrial-strength systems
- Mix monolithic and modular systems modelling
- Easily build systems with mixed symbolic messaging and binary streams
- Dynamic, built-in Quality-of-Service system for communicative and runtime events
- Freely mix publish-subscribe and client-server architectures





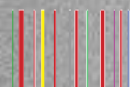
# Why Psyclone 2.0 ?

- Build and test modular configurations more quickly
- Build complex, robust systems more quickly
- Reconfigure faster even the most complex systems
- Monitor and interact with the whole system at runtime
- Connect modules from a library of modules supporting numerous services
- Write custom plug-ins for monitoring and controlling the system with ease
- Redirect streamed audio and video using publish-subscribe mechanisms
- Modify the system configuration at run-time without downtime



# Psyclone 2.0 Delivers:

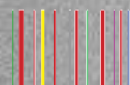
- Robust Autonomy
- Scalability
- Multi-Platform Compatibility
- Faster Development
- Simpler Testing
- Maximum Integration
- Reliable Solutions
- Code infusion for C++



# Psyclone 2.0 Key Features

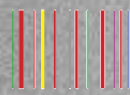
- Distributed (multiple computer) module execution
- Central management of system configuration
- Web-based run-time monitoring of distributed systems
- Near-instantaneous on-the-fly reprogramming of global data flows
- 60k+ computer nodes running Windows and Linux 32 and 64 bit
- 1M+ modules in a single system
- Initiated, configured and managed from a single point of access
- Multiple simultaneous networks with automatic firewall detection





# Psyclone 2.0 Nodes & SDK

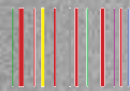
- Distributed system on large computer cluster
- One Psyclone Node on each computer in the system
- Modules created in C++ using the CMSDK
- Currently Windows and Linux 32/64-bit supported, others may follow
- Currently C++ is supported, Java, scripted modules and others may follow



# Single Config File (PsySpec)

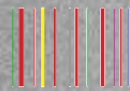
- A single PsySpec XML file sets up the whole system regardless of size
- The PsySpec specifies the nodes to use for the system
- A node can have one or more processing spaces
- A node can have many modules running one or more spaces
- A node can have many Whiteboards and Catalogs running one or more spaces
- A node can have many Feeds, Interfaces and Services
- A module can specify one or more active contexts
- Module subscriptions are specified for each context
- A subscription lists triggers, cranks, posts and retrieves





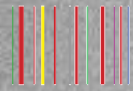
# Psyclone 2.0 Modules

- Modules produce and subscribe to data messages by their type
- A module resides inside a Space on a Node in the system
- Some modules can dynamically migrate to other nodes
- Modules are triggered when new data is available to process
- Modules can sleep or do background processing in-between triggers



# Psyclone 2.0 External Modules

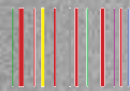
- External modules are created inside third-party software applications
- They allow other software to participate in a distributed system
- They can connect and disconnect according to their own requirements
- The other software merely needs to include the CMSDK library
- All local communication goes via shared memory



# Psyclone 2.0 Module Parameters

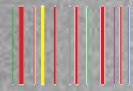
- All modules can have multiple parameters
- They can be numerical, text or collections
- They can specify a range interval and step size
- Parameters are like knobs that can be tweaked by other modules





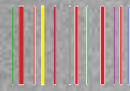
# Psyclone 2.0 Whiteboards

- Whiteboards are indexers of messages in the message banks
- They can index on any field in the messages
- Modules can query and retrieve sets of data using complex queries
- Queries can include filter such as time, count, etc.



# Psyclone 2.0 Catalogs

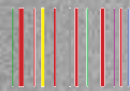
- Catalogs stores and manages data
- Data can be from internal or external sources
- Modules can query and retrieve sets of data using complex queries
- Queries can include filter such as time, count, etc.



# Psyclone 2.0 Data

- Data flows as discrete messages in the system
- All local data flows via shared memory
- All other data flows via network (TCP/UDP, others may follow)
- Messages remain in memory banks until their time-to-live expires
- Data flows based on the currently active subscriptions
- Subscriptions change near-instantaneously based on active system contexts





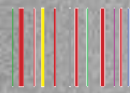
# Psyclone 2.0 Message Types

- Message types are textual names separated by a period:

*aaa.bbb.ccc.ddd*

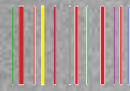
- Up to 16 subtypes can be used in a single type
- Wildcards can be used for matching:

*aaa.bbb.\*.ddd*



# Psyclone 2.0 Topics

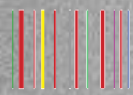
- A *topic* is a string attached to a message (if created by human) or a numerical value (if auto-generated)
- Topics are global and can be used to segment processing into clusters where the group only deals with a particular data segment
- A module can attach a topic to a series of messages
- This can be to group data concerning a particular object or event
- A module can be asked only to be triggered by its own topics
- Topics remain attached to message chains



# Psyclone 2.0 Signals

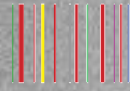
- Signals are faster, low-level messaging
- They are broadcast to modules currently listening for the signal type
- They can be used to synchronise processing of many modules





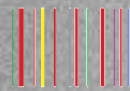
# Psyclone 2.0 Processing Spaces

- Each node can have many processing spaces
- Each space is a separate process, shielding the node and other spaces from crashes
- A crashed space can be restarted automatically
- All communication flows via the node's shared memory



# Psyclone 2.0 Subscriptions

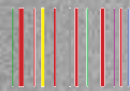
- A subscription consists of one or more named trigger conditions
- Each trigger has a processing statement (“*crank*”) about which piece of code to run
- Each trigger can have one or more named posts for directing output data
- Each trigger can have one or more retrieves specifying additional data from memory banks



# Psyclone 2.0 Contexts

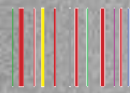
- Contexts are system-wide named states which can be active or inactive
- They are textual names separated by period:  
*aaa.bbb.ccc.ddd* (like message types)
- Subscriptions are active when their context is active
- Only one active context per root (*aaa.x.x.x.x*)
- Many simultaneously active roots (*aaa.x* and *bbb.y*, etc.)
- Up to 16 subcontexts can be used in a single context





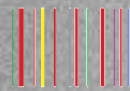
# Psyclone 2.0 Interfaces

- Interfaces link internal resources with external users
- Can be for input, output or bidirectional data
- Examples are HTTP, telnet, native messaging
- More protocols may be added in the future
- Interfaces links with services which says what do to with the data



# Psyclone 2.0 PsyProbe 2.0

- PsyProbe is the web interface for monitoring and interacting with a system
- It can live monitor all activities and data flowing through the system
- Great for debugging systems as they are being developed
- Uses Web 2.0 technologies that should run in all browsers
- A mobile interface will be added in the future
- An XML-based API can be used for third-party system monitoring
- Third-party plugins for individual modules allows for custom interaction

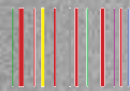


# Psyclone 2.0 Services

*(partly implemented in current beta)*

- Links interfaces with internal data messages
- Allows custom external data to be translated to internal messages and vice versa
- Can set off large processing chains and be triggered with the results
- Each chain will be marked with a topic to allow many distributed services to run

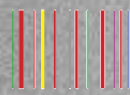




# Psyclone 2.0 Module Data

*(partly implemented in current beta)*

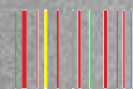
- All modules can choose to save their local data to the node shared memory
- This will enable them to be restarted or migrated to other nodes
- If a module space crashes modules can be automatically recreated and continue working



# Psyclone 2.0 Performance Monitoring

*(partly implemented in current beta)*

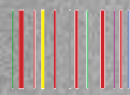
- Continuous full system performance monitoring
- Viewable by the operator
- Used to dynamically change system parameters to optimise runtime performance
- Will be used for automatic load balancing



# Psyclone 2.0

Features to be included in  
future releases

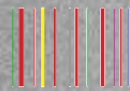




# Psyclone 2.0 Subscription Triggers

(future feature – not yet implemented)

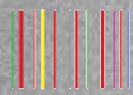
- A trigger is based on message types, optionally with wildcards
- A trigger can specify time filters (message maxage)
- A trigger can specify additional *key=value* filters
- A trigger can specify a topic
- A trigger group can wait for many triggers with Boolean conditions on when to trigger



# Psyclone 2.0 Feeds

(future feature – not yet implemented)

- Feeds produces a steady stream of data
- Input can be from internal or external sources
- Output can be messages for internal consumption
- Or streams for external consumption (RSS, files, media)

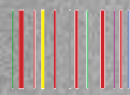


# Psyclone 2.0 Scripted Modules

(future feature – not yet implemented)

- Some modules can be scripted in the PsySpec directly
- The scripting language will most likely be AngelScript
- Other scripting language can be supported in the future

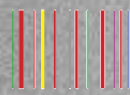




# Psyclone 2.0 Topic Module Instances

(future feature – not yet implemented)

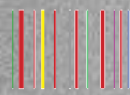
- One or more modules can be auto-spawned upon creation of a *topic*
- This can be used for dynamic event handling, e.g. handling a specific external resource connecting
- The modules are automatically destroyed when the topic is destroyed, e.g. when the external resource has finished and disconnects



# Psyclone 2.0 Multiple Module Instances

(future feature – not yet implemented)

- One module definition can spawn any number of copies
- Each copy will have a unique name and exist independently
- Parameter values can be randomised by range

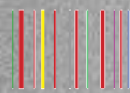


# Psyclone 2.0 Android & iOS SDKs

(future feature – not yet implemented)

- SDKs for Android and IOS will be added in the future
- This will enable mobile apps to participate in the running systems

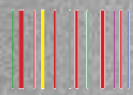




# Psyclone 2.0 Automatic Load Balancing

(future feature – not yet implemented)

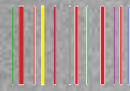
- Automatically spawned modules can be placed depending on available resources
- Existing modules can be migrated to nodes with more available resources
- Custom load balancing policies can be created by the operator



# Psyclone 2.0 Recording & Playback

(future feature – not yet implemented)

- Module input and outputs can be recorded for later simulation and playback
- Individual modules can be debugged in isolation
- Individual modules can be simulated initially during development



# Psyclone 2.0 Licensing

- Open-Source with dual licensing
- LGPL for personal and commercial use
- Commercial licence for non-LGPL use and support