



NetDog Massive Multiplayer Networks

Networking Solution For Massively Multiplayer Online Games

Technology Overview



NetDog Overview

NetDog is a networking engine for Massively Multiplayer games that executes time synchronized events from disparate users across highly scalable networks. Created specifically as networking middleware, NetDog easily integrates into games, connecting to all game components seamlessly, providing the best-in-breed solution for any type of online game. NetDog is cross-platform, letting gamers on PC, Mac, Linux or iPhone play together in the same game.

NetDog is built on the following core components, operating transparently with minimal need for user configuration:

- Network and Socket channels (TCP, UDP)
- Connection-level options (encryption, clock sync, reliable UDP)
- Event filtering system (validation)
- Event callback system (event handling)
- Object system
- Transparent optimizations (message aggregation, fragmentation)
- High-level features (RPCs, File Transfers, etc.)

NetDog licensees not only get state-of-the-art game networking technology, but also a reliable technology partner who provides competent support and is ready to help with any integration issues.

NetDog Benefits

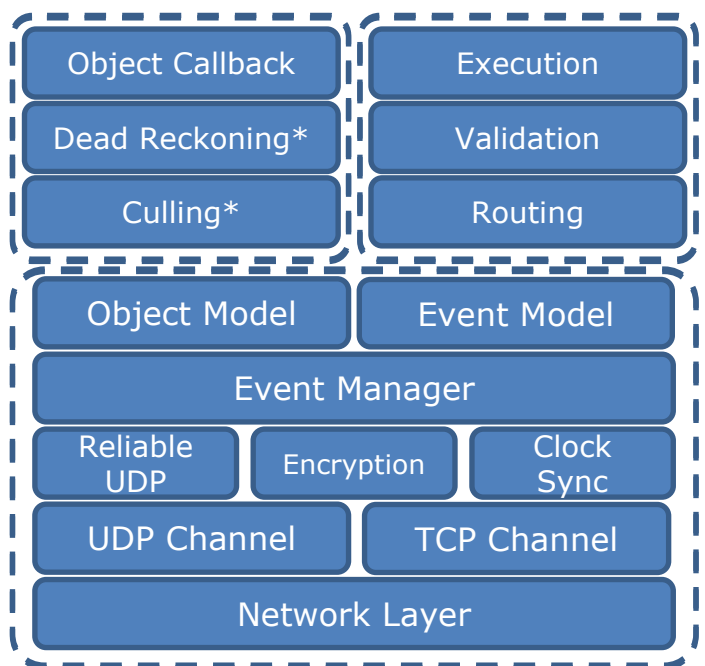
- **Highly scalable** to create huge worlds that hold up under heavy load
- **High performance** allowing you to create fast-action, FPS-style games
- **Easy to use and easy to integrate** reducing time to market, and letting developers focus on building great games
- **Flexible** making it compatible with most commercial game engines

NetDog Technology

At NetDog's core is its messaging system that transparently handles reliability, encryption, and clock synchronization on a per-connection basis. NetDog messages are called "Events" and are handled by an Event Manager. Events can be used directly in the Event Model, or the Object Model can be used instead (or in addition).

The object model provides an object-centric view of data and transparently handles object updates for you, applying culling and dead reckoning to object updates. Finally, the user can install an Object Callback which NetDog will call whenever receiving object-related events.

The Event Model is simple yet powerful, with configurable routing which defaults to sending messages to all clients. Validation callbacks allow the server to determine which messages to execute. Finally, event execution is made possible by user-installable callbacks.



NetDog is built for highly scalable performance from the ground up, providing minimal overhead to game events, optimizing networks with:

- Low-level networking stack that is optimized for scalability and performance
- Packet overheads among the smallest in the industry
- UDP Protocol provides low latency, with reliable UDP also available

* This feature will be available in a near-future NetDog release



Performance Tools

NetDog provides optimization tools that improve network performance, allowing for fast action in even the most demanding of applications:

- Automatic message fragmentation, allows sending of large events/RPCs, up to 4GB in size.
- Automatic message aggregation, allows multiple small events/RPCs to transparently combine into a single network packet, reducing network load.
- Optimized message pipeline, allows concurrent execution of events through the use of "tasklets".

High Level Functions

NetDog offers several high-level functions, saving you the trouble of writing these tools yourself:

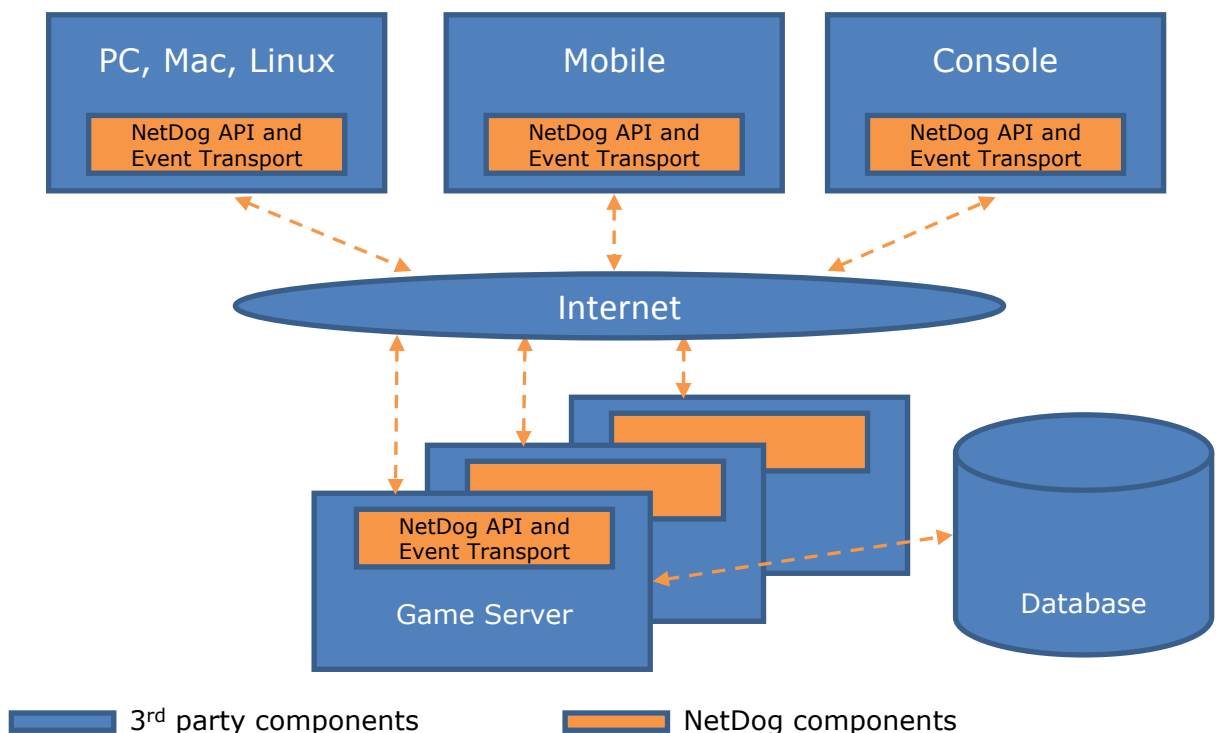
- Remote procedure calls make writing applications faster and more intuitive.
- File transfer functions make it incredibly easy to transfer entire files of virtually any size (subject to OS limitations).
- Event system transparently breaks down large events for network transport, allowing events up to 4GB in size.
- Object system stores and synchronizes game data in an intuitive, object-oriented fashion.

Integrating with NetDog

NetDog's API interface provides robust, yet flexible access to a complete networking solution for massively multiplayer games – while making it possible to integrate NetDog into an existing game in as little as a single day.

NetDog integrates into game engines from various platforms using intuitive bindings that incorporate all other tools necessary to create rich, robust multiplayer online games. In addition, NetDog has established partnerships and integrations with the following middleware products and platforms:

- Physics engines (Ageia/PhysX)
- Game engines (Trinigy, Irrlicht, Unity*)
- Platforms (Windows, Linux, OSX, iPhone*, Console forthcoming)
- * Support currently limited





How it works:

NetDog is designed to meet a wide range of needs, and integrate into a variety of environments. Thus, implementation will vary depending on the type of game, and complexity of the game environment. However the following are the basics that will be common across all games.

Basic Implementation

Basic integration of NetDog with a game engine can be performed with just a few function calls, quickly creating a multiplayer world from any stand-alone game. In addition, NetDog is integrated with several 3rd party engines, and includes out-of-the-box tools for integration, further accelerating development. Implementation is also the same across platforms, so no platform-specific work is needed.

1. Create connections

NetDog manages connections and sockets transparently. All socket programming, low level networking and messaging are handled in one function call that allows for customization through a flags for reliable UDP, encryption, tunneling and other connection functions.

1 Create connections

```
Server:      createChannel ()
             becomeLoginAcceptor ()
Client:     loginToHost ()
```



2. Map in-game events and objects to NetDog events and objects

NetDog's Event and Object Models are based upon flexible data structures that make it simple to map in-game processes to NetDog events and objects. Events and objects are handled using user-installed callback functions. NetDog has example integrations for several 3rd party engines, and our support team will work with you to integrate into any new engines.

2 Map Events and Objects

```
registerEventType (...)
registerObjectType (...)
```



3. Execute event processing loop

The Event Loop performs time-stamped execution of all NetDog functions, ensuring that all nodes in the game are synchronized. Messages are sent from client nodes to their corresponding servers where they are validated, routed, executed, and sent back to the client nodes for local processing (e.g., local physics, rendering, etc.).

3 Execute Event Loop

```
mainLoop (...)
```



Contact Information

PX Interactive knows that each game is different and will require different networking support. NetDog is highly customizable, and can meet these ever-changing needs. Contact us to find out how NetDog can help you quickly create networks for your multiplayer worlds.

Support contact

sales@netdognetworks.com

+1 415 692 6757

<http://www.netdognetworks.com>