

# Architecture

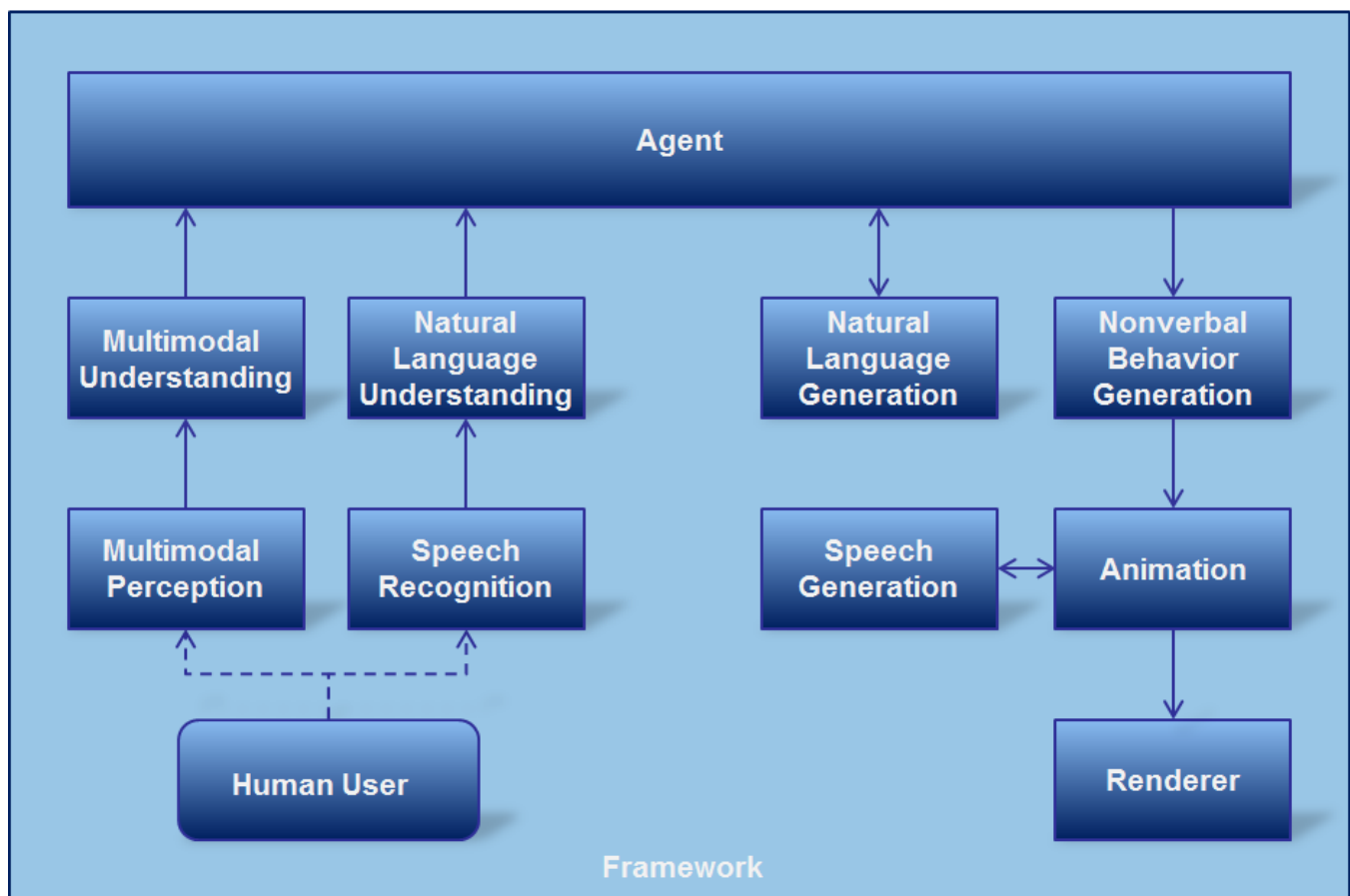
- [Virtual Human Architecture](#)
- [Virtual Human Toolkit Implementation](#)
- [Data Models](#)

## Virtual Human Architecture

The Virtual Human Toolkit is based on the ICT Virtual Human Architecture. This architecture defines at an abstract level what modules are needed to realize a virtual human, and how these modules interact. The basic functionality of each module as well as its interface is well-defined, but its actual implementation falls outside the scope of the architecture. The architecture dictates the implementation of a distributed system where communication is mostly realized using [message passing](#). This allows for multiple implementations of a certain module and simple substitution of one implementation for another during run-time. It also allows for distributed systems, where different modules run on separate computers.

ICT has developed a general framework, consisting of libraries, tools and methods that serve to support relatively rapid development of new modules. Using this framework, ICT and its partners have developed a variety of modules, both in the context of basic research as well as more applied projects. The Toolkit provides some of the modules that have been transitioned from this research.

Please see below for a high level Virtual Human Architecture.



## Virtual Human Toolkit Implementation

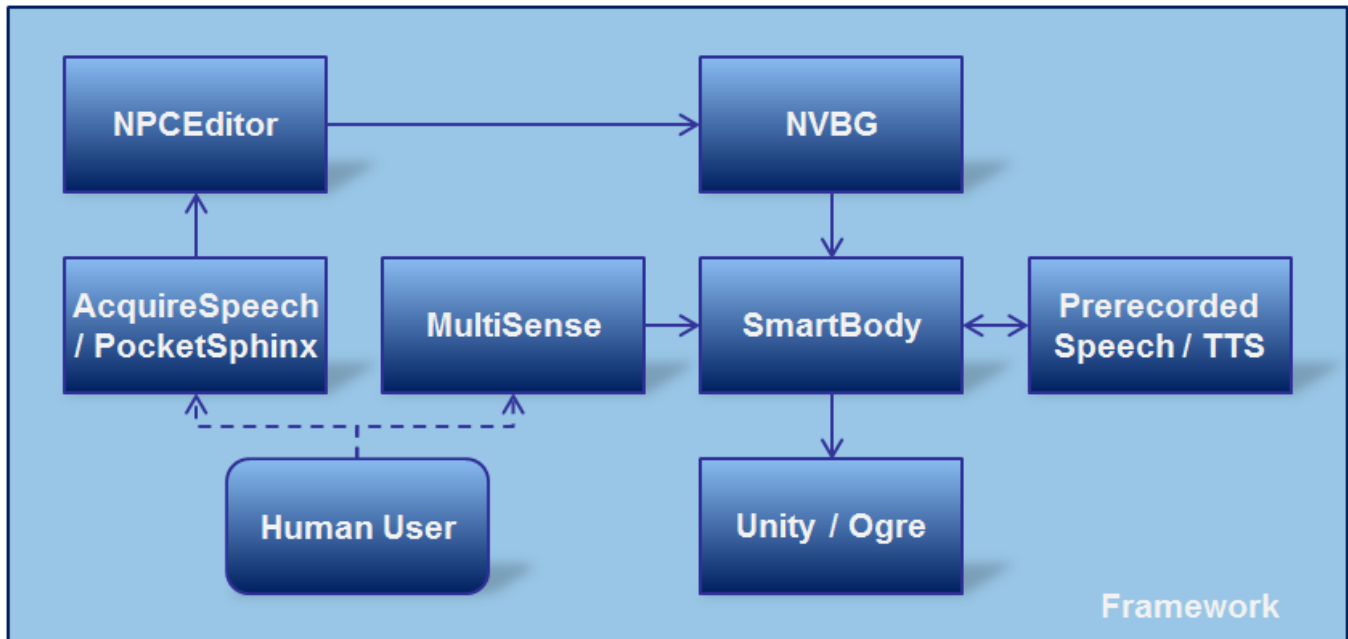
The Virtual Human Toolkit is a set of components (modules, tools and libraries) that implements one possible instantiation of the Virtual Human Architecture. It has the following main modules:

- [NPCEditor](#), a statistical text classifier which matches novel input (a question the user asks) to authored output (a line the characters speak).
- [SmartBody](#) (SB), a character animation platform that provides locomotion, steering, object manipulation, lip syncing, gazing and nonverbal behavior in real time through the Behavior Markup Language (BML).
- [NonVerbal Behavior Generator](#) (NVBG), a rule based system that takes a character utterance as input and a nonverbal behavior schedule (gestures, head nods, etc.) in the form of BML as output.
- [MultiSense](#), a perception framework that enables multiple sensing and understanding modules to inter-operate simultaneously, broadcasting data through the Perception Markup Language (PML).
- [Unity](#), a proprietary game engine. The Toolkit only contains the executable, but you can download the free version of Unity or purchase Unity Pro from [their website](#). The Toolkit includes Ogre as an open source example on how to integrate SmartBody with a renderer.
- [PocketSphinx](#), an open source speech recognition engine. In the Toolkit, PocketSphinx is the speech server for our [AcquireSpeech](#) client.

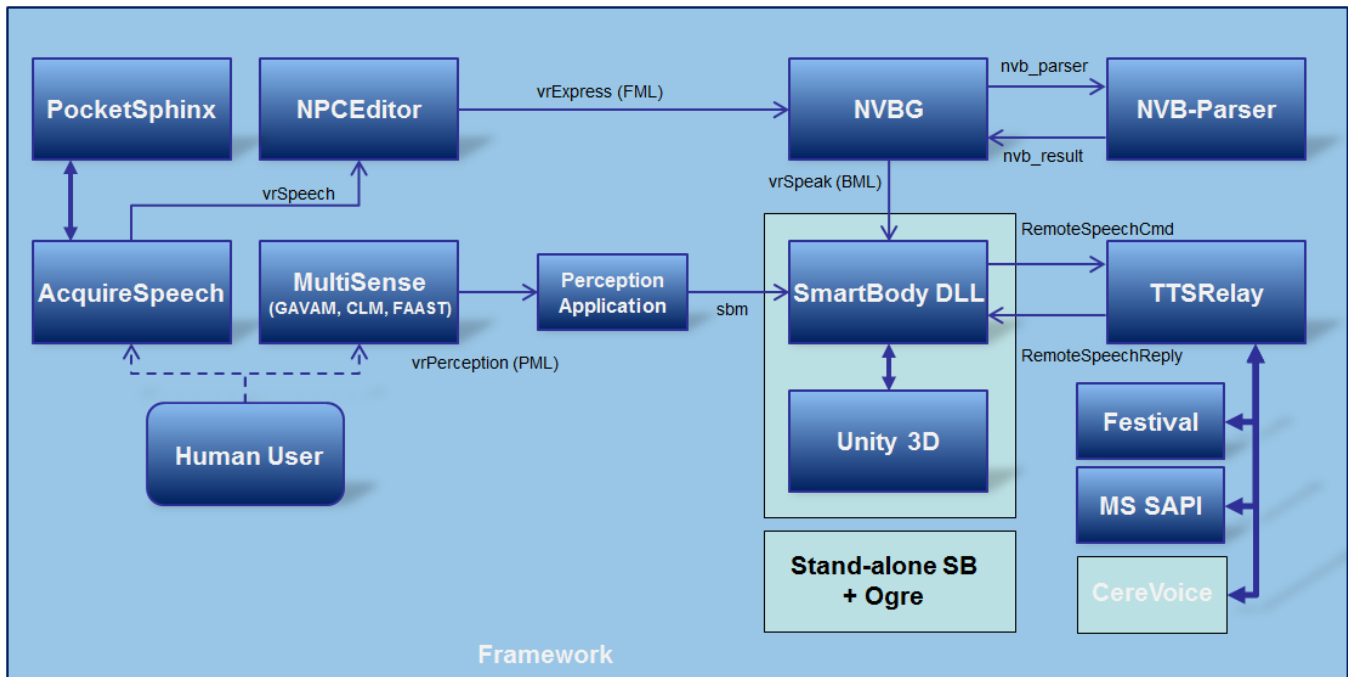
- [Text-to-speech engines](#), including Festival and MS SAPI.

For a complete overview of all the modules, tools and libraries, please see the [Components](#) section.

The figure below shows a high level overview of the Toolkit modules. Instead of a full-fledged agent, the NPCEditor is handling all verbal in- and outputs. Note that since MultiSense (perception) is currently included as a basic proof-of-concept, it is directly communicating to SmartBody. A deeper integration with the system would require MultiSense to communicate with the NPCEditor, a dialogue manager and/or the NVBG instead.

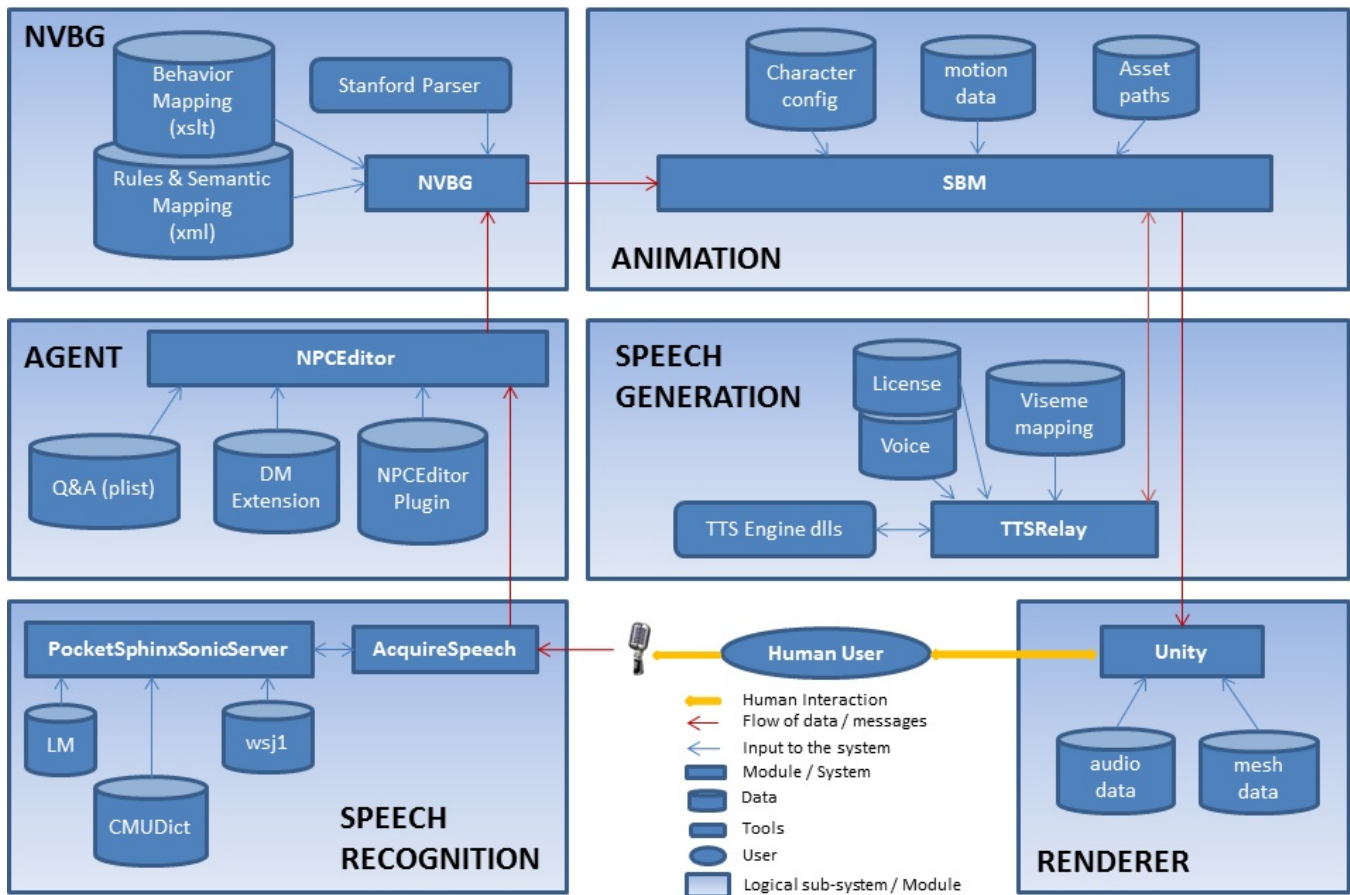


A more detailed overview is depicted below. The bold arrows indicate a direct link between modules, either TCP/IP or included DLL. All other links show what [messages](#) are passing between modules; see [Virtual Human Messaging](#) for more details.



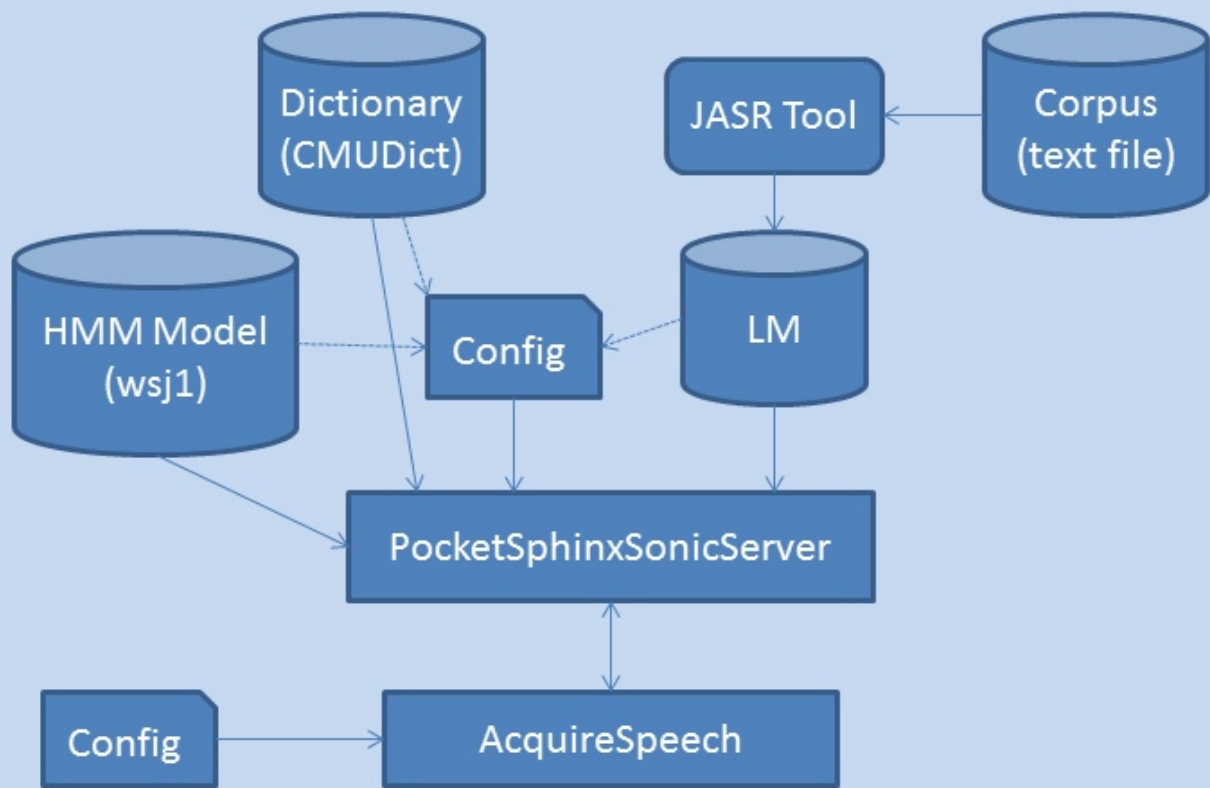
## Data Models

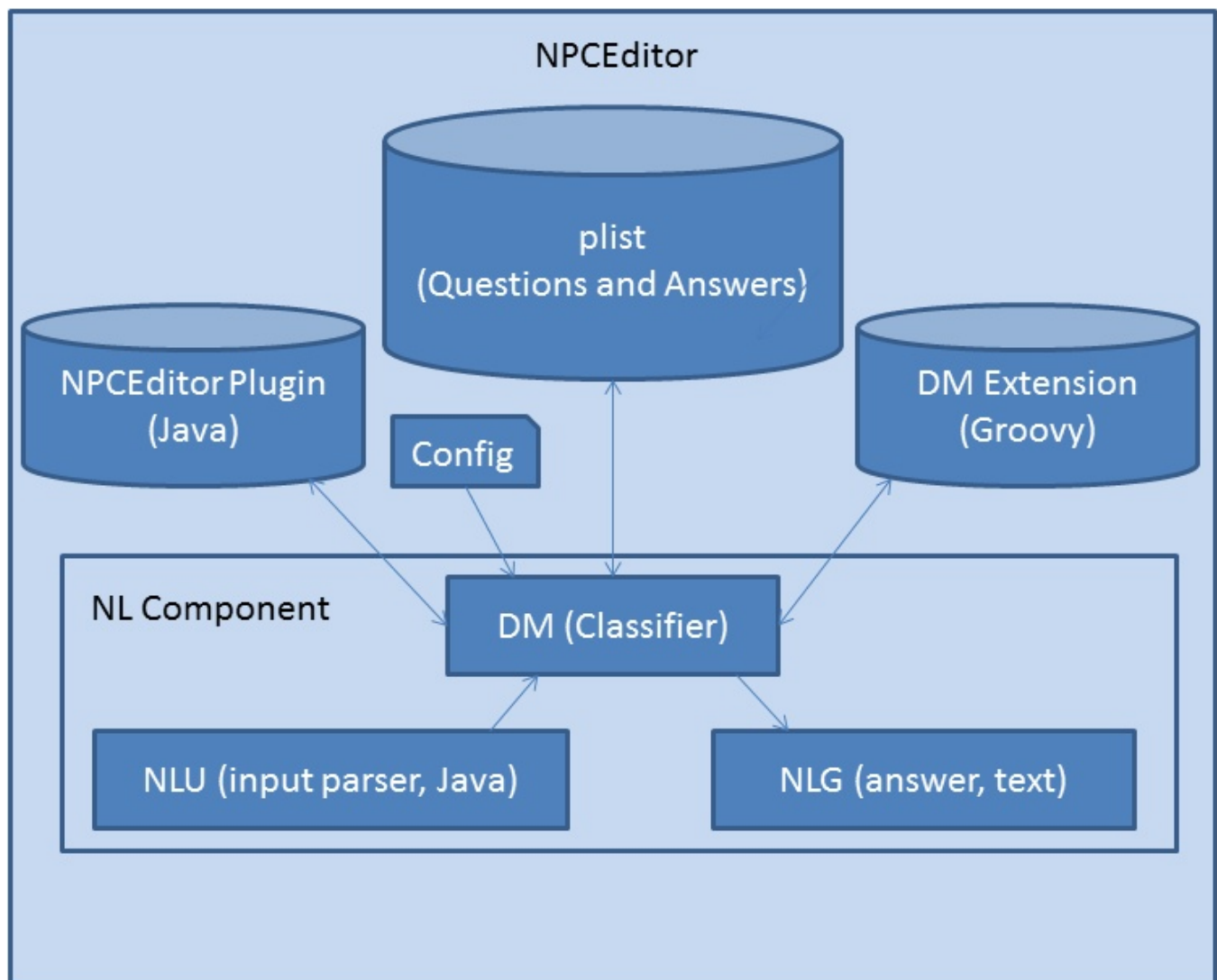
Virtual Human Toolkit:



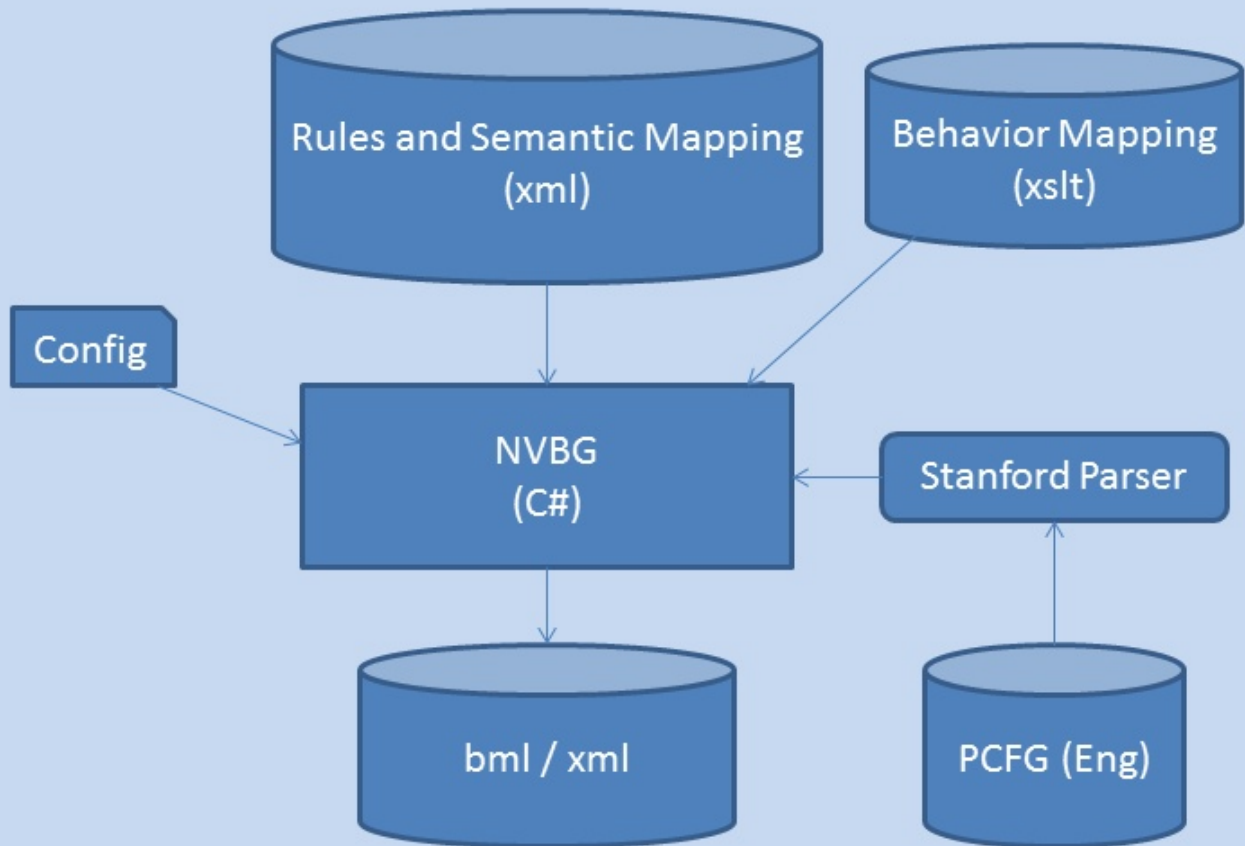
Individual modules:

## Speech Recognition

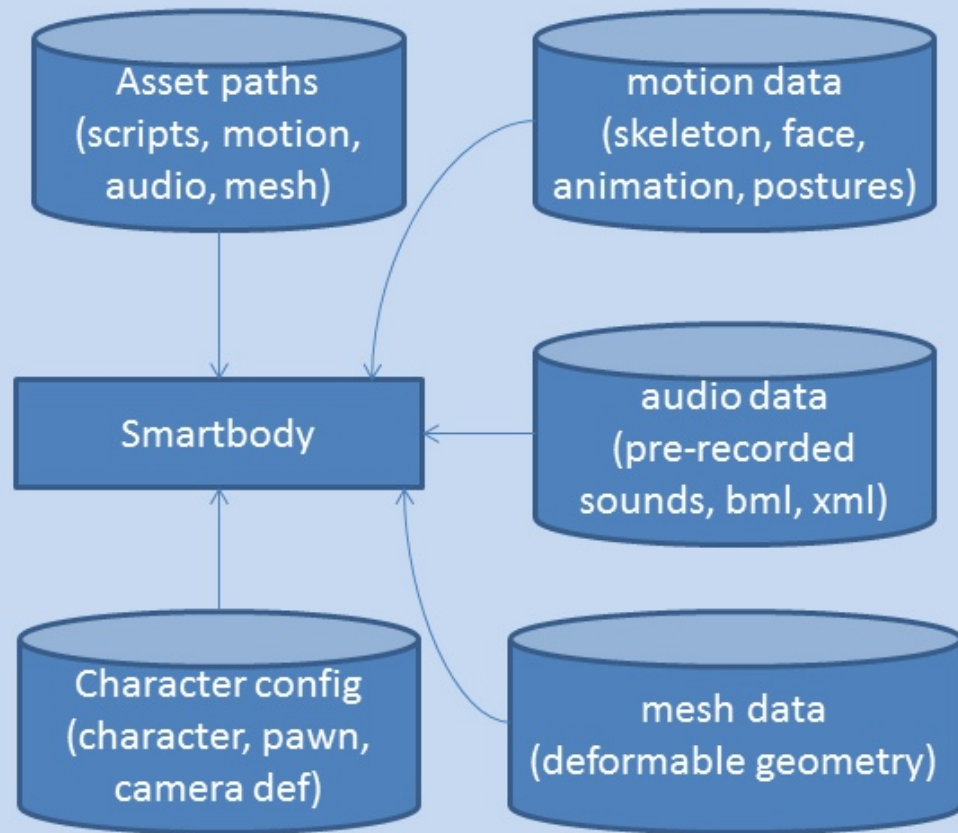




## Non-Verbal Behavior Generator (C#)



## Animation (smartbody)



## Text-to-Speech

