**Channel Creation**

NamePipeChannelCredentials() : Provides creation of channel which support namedpipe operations on it.

Used with CreateChannel()

Diffproc\_channel\_create.cc: Provide channel implementation functions for Name Pipe operations

Named pipe uses CLIENT\_DIRECT\_CHANNEL, which dodges other implementation like creation load balancers, resolvers, and other filters.

It requires transport to be setup before channel creation. So, we initialize transport for client section here.

1. So, while initializing channel, we need to take of two things.
   1. Channel arguments are mandatory at this point, where authority is required, so we provide default authority to satisfy this situation.
   2. To create an endpoint at client end which tries to connect to server endpoint using namedpipe. Once the connection is established it calls on\_done callback, which creates a transport.
2. Now once channel is setup, channel is created.

**Point to note here is we need an Endpoint when we create transport, to make the wire protocol transmit data over this endpoint.**

**Server Channel Creation**

There is no such thing as channel at server end, it basically accepts connection from client which forms a channel with client.

1. So, for that operation to be performed, we must create named pipe and wait for incoming client connections.

namedpipe\_server\_windows.cc

This class provides all the necessary functions, which creates a new namedpipe instance, waits for new incoming client connections.

**Start\_accept\_locked()** function provides feasibility for creating new named pipe instances and append them to a LinkedList data structure, once a client is accepted, the next handle will be serviced to accept a new client. This process continues until the thread is ended.

**Threading Model (Connection thread) ---- namedpipe\_thread.cc**

So the named pipe provides a connection thread model, where a new pipe instance is created and waits on for a incoming client connection using **ConnectNamedPipe().** Once a connection is accepted here, it calls a callback function on\_accept which is function from **namedpipe\_server\_windows.cc**

**Transport Creation**

Once the callback function is called, it creates an endpoint – Adding pipehandle on which connection is accepted to a grpc\_namedpipe struct and Initializing endpoint vtable functions, standard set of functions like {Read, Write, shutdown, destroy etc. } and returns base pointer of that structure.

**Point to note here is we need an Endpoint when we create transport, to make the wire protocol transmit data over this endpoint.**

**Server\_utils.cc**

After the endpoint is created in on\_accept() function, it calls onc\_accept function from server\_utils.cc

At this level, transport is created with endpoint to which client is connected,

Now it calls transport function to perform first read.