idx	**DIMENSIONS**	(counts)		
0	listenerCnt::Int	(Ln)		
1	requestCnt::Int	(Rn)		
	HEARTBEATS			
2	status:Int	heartbeat::Time	initialization:Time	**(watchdog status & heartbeat)**
5	status::Int	heartbeat::Time	initialization:Time	**(server status & heartbeat)**
8	listener[1]:hash	heartbeat::Time	requestReservation::Time	reservationAcknowledgement::Time
…				
8+(4*(Ln-1))	listener[n]::Hash	heartbeat::Time	requestReservation::Time	reservationAcknowledgement::Time
	REQUESTS			
8+(4*Ln)	Listener[x]::Hash	MD5-of-Req#1	Response::Time	**(request status, request, and timesrtamp of completion)**
…				
8+(4*Ln)+(3*(Rn-1))	Listener[x]::Hash	MD5-of-Req#N	Response::Time	**(request status, request, and timesrtamp of completion)**

KEY:

Time:	**TimeStamp**	**UInt64**
Int:	**Integer**	**UInt64**
Hash:	**Hash64**	**UInt64**

To make a request:

1) Listener updates its requestReservationTime
2) Watchdog looks for an available request slot **
3) Watchdog Identifies an available slot with a listener hash
4) Listener finds slot identified for its usage
5) Listener writes request where server can see it
6) Listener does MD5 hash of request and updates its slot
6) Server sees a request slot, reads the request, and services it
7) Server writes the response where the listeners can see it
8) Server updates its request slot with the response time
9) Client sees the response time and reads the request
10) Watchdog looks for listener heartbeats that are greater than the request time

**(where listener heartbeats all exceed the response time by at least 1/2 second)