The overall view has the combination of different design principles and patterns to solve issues that aroused during the global analysis. Some of them with the reasons why they have been chosen are listed below:

- Multi-Layered Architecture:To divide the entire Architecture in higher groups so that it becomes easier to identify the place for modification and improvement whenever necessary. For example :If there is an requirement to changes the message given to the client,we can make direct change to Data layer.
- Distributed System: To balance load.
- Service layer: To be able to integrate with third party components seamlessly.
- Module based: To decrease cohesion so that change at once place doesn't affect others.
- Object oriented: For reusability of the code.
- Single Responsibility principle(SRP) : For maintainability .

Telephone layer:

This product can get input from different sources . For e.g : Mobile,PSTN(Public switch telephone network) etc. It is then passed to CTI(Common telephone Interface). CTI is responsible for routing the call to Business layer.

Business layer:

The message from CTI is received by service interface . This interface accepts input which follows certain standards and are in certain format. After this they are passed through one of many DTMF(Dual Tone Multiple Frequency) . DTMF is responsible to determine if the input by the user is from tone or not.If the input is from tone, it goes to processor and interpreter else it goes to one of many ASR(Automatic Speech Recognition) through service interface and then to processor and interpreter.ASR is responsible to translate speech to text. Processor and interpreter has the logic to manipulate data and act accordingly. For example: If the user is looking for menu option, it should directly move towards the menu . During this this is no need to check the weather information.

After the information is processed it has to be translated to speech so that user can get a reply. TTS(Text to speech) is responsible for this task.

Data layer:

It consist of everything from where we can access information. It can be from local sources like local database or from other applications etc.To facilate the the transfer of infomation between these third party applications service interface layer is used.
Error handling:

Error are handled by the product in a centralised way. Error can come up when the user ask for the weather information of the city whose information are not present or by entering wrong city

code. In this case we might ask the user to enter the correct code or end the call. In any case our architecture looks up the information in data layer through the business layer .

Alternatives to the design

1. Distributed architecture can be replace by centralised powerful server. It makes much sense when the product doesn't have number of user. Our product doesn't use it because we think it is very likely that the number of user will increase as day passes by.

2.The product would have been much more efficient without the use of interfaces, module based architecture ,SRP principles etc. Tightly coupled system are the the most efficient systems . But it is very hard and requires a lot of effort to accommodate change in this system. changes are inevitable and on top of that ASR(Automatic Speech recognition) has not yet reached its maturity and it is bound to change more than others.