

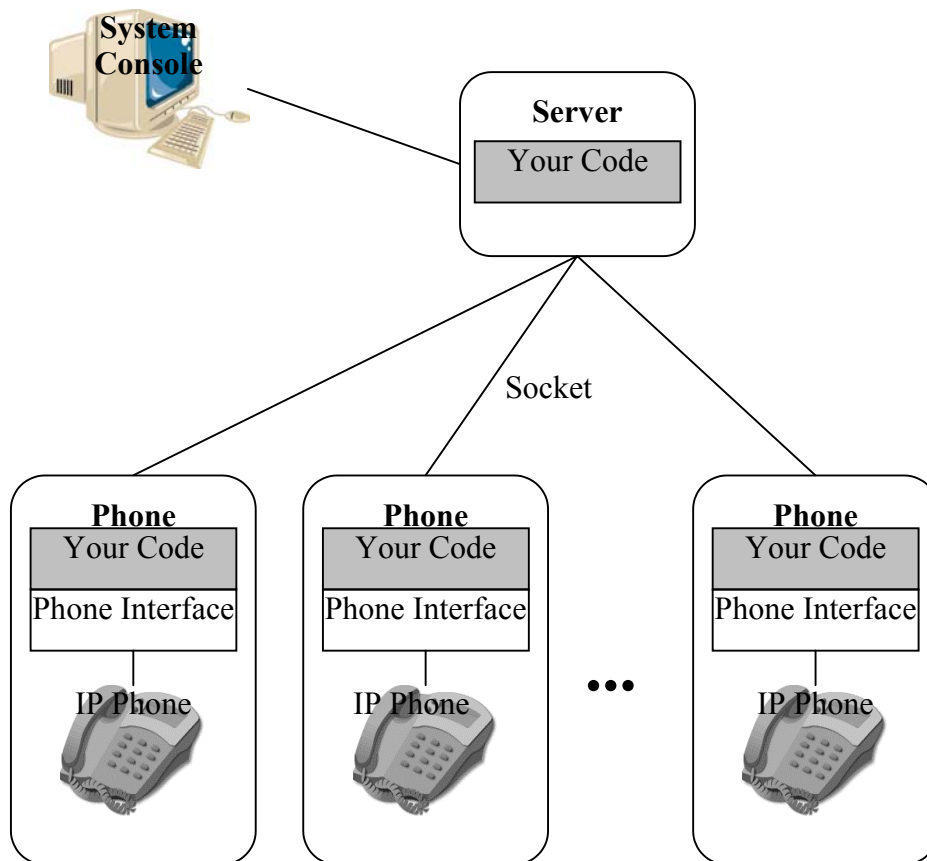
# Software Tender for Voice over IP Telephony SuperTel Incorporated

## 1 Introduction

The following sections together with an accompanying *hardware interface description* (HID) for SuperTel's new IP phone comprise a preliminary requirements definition for a planned new *entirely private branch exchange* (EPBX) to be marked by SuperTel Incorporated. Bid selection will be based on a detailed design proposal for the control and server software.

## 2 System Configuration

The following figure specifies the overall system configuration for a customer installation of SuperTel's new EPBX. Note that shaded areas correspond to the tendered software.



For a given installation, the system consists of a Server and several Phone Subsystems. The Server consists of one or more processes and possibly a database. Each Phone Subsystem uses the API described in the HID document to interface with a single

phone.

There is complete freedom to distribute or centralize the responsibilities of the system over the Server processes and the Phone Subsystems. However, communication between the Server and each Phone must be based on internet protocols such as TCP. This will enable SuperTel to use COTS IP networking hardware components together with SuperTel's new IP phones and commodity PCs for a customer installation.

### **3 Use Cases: Basic Call Processing**

Users should be able to send and receive calls. Sending and receiving calls is accomplished by having the system monitor each phone for events, e.g., off-hook, numbers being dialed, etc., and in the case of a successful call, having the system establish an audio connection.

As an example, the system allows a user, Alice, to pick up her phone, and dial Bob's phone number. Bob's phone then starts ringing, causing him to answer his phone. At this point, an audio connection is established so that Alice and Bob can speak to one another. The audio connection is terminated when either Alice or Bob hangs up.

Note that the above scenario describes a successful call. The system must also handle scenarios that deviate from this sequence, e.g., the caller dials an invalid phone number, the call recipient's line is busy, the caller hangs up before the connection is established, the caller is on a phone that cannot originate calls, etc. The system must also handle circumstances in which resources are not available for the call, e.g., the maximum number of calls has been reached.

### **4 Dialing Plan**

To facilitate basic call processing the system needs to construct and maintain a mapping from dialed numbers to IP addresses. The software uses this mapping to translate the number dialed by a caller, e.g., the digits dialed by Alice, into the IP address of the call recipient, i.e., Bob.

Note that this mapping needs to be accessible to the software that does Call Processing as well as the software that handles Administration. Due consideration must be given to the potential conflicts this may cause, e.g., should a request from the software that does Call Processing have higher priority than a request from the software that handles Administration? Should caching be used somewhere?

Phone numbers, also called *extensions*, consist of four digits. The system is free to make certain numbers special if appropriate. For example, if users are to be allowed access to advanced features, they can specify a certain prefix as the indicator for the advanced feature. (A concrete example from the public phone system is dialing 1 to call long-distance. A side effect of this special prefix is that no phone numbers can begin with a 1.)

### **5 System Console**

The System Console is a Graphical User Interface used by a client's personnel to monitor and control a network of phones. These personnel, called Administrators, must login to the System Console using a login name and a password. It may be assumed that every Administrator has access to all the features of the System Console.

It may also be assumed that at most one Administrator accesses the system at a time.

Regardless, some functions, e.g., finding an available phone number, should nevertheless be atomic. (This will make it easier to extend the system to accommodate multiple Administrators.)

The software is responsible for checking the Administrator's input to ensure that it does not cause the system's database to become inconsistent. For example, requests to map a single phone number to two different IP addresses should not be allowed.

Additional features of the System Console are described, sometimes implicitly, in the following subsections.

## ***5.1 User Accounts***

When a new phone with a given IP address is added to the system, an Administrator must find an unassigned phone number and associate the phone number with the IP address.

Each user account has some combination of the following privileges:

- Call origination.
- Call answering.

Note that implementing these privileges using a filtering mechanism enables more powerful features to eventually be deployed. For example, the CEO of a company might be able to define a filter to ignore incoming calls from everyone except high ranking executives and their secretaries.

To cancel a user's telephone service, an Administrator must disassociate the IP address from the phone number, and disassociate both from the user's account, thereby making the IP address and phone number available to future users. User records should be retained indefinitely.

## ***5.2 System Maintenance***

The System Console can display hardware status information, such as which phones have been allocated to a user, which devices, if any, are out of service, as well as any other information deemed relevant.

An Administrator may request hardware tests at any time. For example, if a user complains that his or her phone does not work, the Administrator may run a connectivity test to verify that the user's phone is connected to the system. If a problem is detected, the Administrator may request that the phone be taken out of service. A detailed design is free to determine how an out of service phone is handled in the system. Similarly, if a new or repaired hardware device is added to the system hardware, the Administrator may request that the device be enabled.

In addition to requests from the Administrator, the System Console must be able to receive error messages, e.g., Hardware Failure. The System Console must alert the Administrator whenever there is an error, e.g., new negative test results, so that the operator can take immediate action, e.g., request additional tests, take the device out of service, or both.

### ***5.2.1 Automatic Hardware Fault Detection***

The system should monitor all phone processes and periodically run tests to verify that the phone is operating normally. For example, a message might be sent to each phone process every thirty seconds. If a phone process does not respond within a reasonable

length of time, then the system can assume that an error has occurred.

If a phone is turned off, unplugged, or suffers a catastrophic software failure then the system must detect this and alert the operator of the failure. The phone software may fail in any number of ways, like any piece of software. The system's automatic hardware fault detection needs to detect only that a phone has become completely unresponsive. To prevent resource leakage, any resources in the system associated with an unresponsive phone must be released.

When the problem with a previously unresponsive phone is corrected, the system will detect the correction and make the phone available for use again, restoring any resources the phone was previously using.

### **5.2.2 Call Processing Reset**

Call processing software tends to become very complicated, largely due to its concurrent nature. If call processing stalls due to some concurrency bug or for any other reason, the system needs to be able to gracefully recover to some default state, that is, without completely stopping the process and restarting it, and without shutting the system down and restarting.

The Administrator must have the ability to reset the call processing for any phone. Depending on the architecture of the system, this ability may mean resetting the call logic in the affected phone process, the server, or both.

Note the difference between Call Processing Reset and the Automatic Hardware Fault Detection. When call processing is reset, the affected process or thread may still be responsive. However if another thread is waiting on an impossible event, the resetting must be able to recover the call processing from the waiting thread.

### **5.2.3 Load Balancing**

The Administrator must be able to set a value for the maximum number of calls allowed in the system. When a user tries to place a call, the system should verify that the maximum number of calls has not been exceeded. A desired but non-essential feature of the system is to display the system load, and to have a way of calculating a suggestion for the Administrator.

It is important to consider when to check or increment this number. That is, should it be incremented when the user picks up the handset, returning a dial tone only if he or she can place a call, or should it be incremented when the system attempts to establish an audio path? The latter might annoy the user whose call is not allowed, i.e., he or she wonders, "Why didn't you tell me before?"

It is also important to consider what happens if a phone crashes during a call or if a phone is unplugged when the system decrements the number of calls.

## **5.3 Billing**

A cost model for telephone calling is required and should depend on the number dialed, the duration of calls, when calls were established, i.e. the time of the day and the day of the week, and the caller's *plan* (defined below).

To facilitate resource utilization, the system must keep a record of every established call. This record indicates who the caller was, so that he or she can be *billed*; the destination phone number; the time and duration of the call; as well as any other information that is necessary.

Each user is sent a *bill* showing all charges incurred during a defined billing period. The System Console can display a user's bill for any *billing period*. The default billing period is the *current billing period*. For each call a bill shows, at a minimum, the number dialed, the day of the week and the time of the day when the call began, the duration of the call, the rate per minute for the call, and the charge for the call; the bill also shows the total charge for all calls and the sum of all charges incurred during the relevant billing period.

At the end of every billing period, the software automatically issues a bill to each user who has an outstanding balance. In addition, a bill is immediately issued to a user if the user's service is cancelled.

An Administrator using the System Console records Bill payments. If a user fails to make a payment, a warning appears on his or her next bill. An Administrator has the authority to suspend or cancel a user's telephone service if the user fails to pay his or her bill.

An Administrator can change amounts charged for calls by adding new billing plans or editing existing plans. The Administrator can also change which plan a user subscribes to. Each billing plan specifies the following.

- The regular charge rate for calls.
- One or more periods, i.e. days of the week and times of the day.
- A discount rate for each of these periods. For simplicity, one may use a percentage of the regular charge rate.

The software is responsible for checking that the dates and times of the discount periods don't overlap. Changes to a user's billing plan become effective immediately.