



VoIP and it's challenges

CONTACT INFORMATION:

phone: +1.301.527.1629

fax: +1.301.527.1690

email: whitepaper@hsc.com

web: www.hsc.com



PROPRIETARY NOTICE

All rights reserved. This publication and its contents are proprietary to Hughes Systique Corporation. No part of this publication may be reproduced in any form or by any means without the written permission of Hughes Systique Corporation, 15245 Shady Grove Road, Suite 330, Rockville, MD 20850.

Copyright © 2006 Hughes Systique Coporation

TABLE OF CONTENTS

<u>SECTION</u>	<u>PAGE</u>
1.0 AN INTRODUCTION TO VOIP AND IT'S CHALLENGES	4
1.1 ABSTRACT	4
1.2 INTRODUCTION.....	4
1.3 A BRIEF OVERVIEW OF HOW VOIP WORKS.....	4
2.0 CHALLENGES FACED IN VOIP DEPLOYMENTS.....	6
2.1 NAT AND FIREWALL TRAVERSAL	6
2.2 QUALITY OF SERVICE (QOS)	7
2.3 TRUST MODELS AND IDENTITY	8
2.4 SPAM OVER INTERNET TELEPHONY	8
2.4.1 Is SPIT a real problem ?.....	8
2.4.2 Why is SPIT a bigger challenge than SPAM ?	9
2.4.3 SPIT in Walled Garden vs. Peer2Peer networks.....	9
2.4.4 Various ways to address SPIT.....	10
3.0 REFERENCES.....	11

1.0 AN INTRODUCTION TO VOIP AND IT'S CHALLENGES

1.1 Abstract

This paper presents a brief introduction on VoIP and discusses some of the most common deployment challenges that are faced. It also discusses the issue of Spam over Internet Telephony (SPIT), its Implications and mechanisms to address this growing concern.

1.2 Introduction

In general terms, VoIP (Voice over IP) refers to a technology domain that specifies protocols which enable users to utilize an IP network for transmission and reception of voice. Specifically, it was originally conceived as a cheaper alternative to dedicated circuit switched lines between calling and called parties, since IP provides a packet based infrastructure that can be re-used by several hosts at the same time. Today, the 'V' in VoIP is a misnomer as the technology has evolved significantly where users can utilize the IP network to transmit and receive voice, video as well as rich data, which is collectively referred as 'Multimedia'.

1.3 A brief overview of how VoIP works

This section provides a brief overview of how VoIP works. The following functionality is needed for establishing an end to end call between two parties (the simplest case):

- A protocol that is responsible for establishing and terminating the call. The primary responsibilities of this protocol are to locate the users, negotiate parameters for the call, setup the call and then terminate the call when a user hangs up. To draw an analogy to human communication, if a chinese speaking individual wishes to communicate with a spanish speaking individual, unless they agree to a common 'language' of interaction, it is impossible for them to communicate. Similarly, the protocol that is used to initiate and terminate this call must be universally understood between the parties involved in the call. As of today, there are two primary protocols responsible for this layer of interfacing – SIP and H.323 with the former having being adopted by most of the next generation network deployments.

A bit of history: H.323 was conceived by the ITU as a protocol that aimed to emulate existing telephony services over a packet network, while SIP was conceived by the IETF. While both are well known standardization bodies, ITU approach the problem from a 'how to fit telephony services into a distributed packet network model' while IETF approached the problem from 'the Internet is the architecture - if it requires that traditional telephony services are mapped to the Internet, then the services need to be rearchitecture to fit the Internet and not the other way around'. After several years of contention, technical differences, and a lot of marketing and hype, SIP has evolved to become the more dominant protocol, even though in terms of deployment, H.323 is still ahead. All new deployments, however are SIP based.

- A protocol which is suited for carrying real-time content. Specifically, when 'media' (as an example, your voice) is transported over an unreliable network (example Internet), a mix of reliability and real-time Delivery needs to be achieved. As an example, if TCP were used over IP, which is a reliable protocol, a single packet loss would result in a protocol stack to wait till it is received or a timeout occurs before the next packet is accepted. From a 'user experience' perspective, would you prefer that you hear your friend's voice with a few 'breaks' (packet loss) or would you prefer that the protocol waits for all packets to be delivered in sequence, which could mean that there are several seconds/minutes of delay depending on how many packets are lost ? A common choice would be to opt for the former for near real-time delivery. However, a purely unreliable protocol is also not acceptable (such as UDP) since for media delivery, there is a need for stream synchronization (for example, if you are hosting a video call, you would want the video to sync with the audio), delay compensation and other media 'alignment' issues. RTP (Real Time Protocol) is the most popular protocol that is used to provide this balance between

connection-oriented protocols such as TCP and connection less protocols such as UDP and is well suited for application payloads containing 'media'.

- Codecs - finally, considering that voice, video and other media streams need to be packetized and sent over a packet network, there is a need to encode the analog signal into a digital signal. In addition, another primary function of the codec is to 'compress' the media so that the amount of bandwidth it takes to transmit the packet(s) over the network reduces, thereby resulting in lesser network congestion and better quality of service. At a very simplistic level, consider this as an analogy to what ZIP does to raw documents - they are compressed so that it takes lesser time to transmit them via email and reduce the chances that the document does not choke the mail server. There are several codecs that are standardized as well as those that are proprietary, each with certain characteristics that make them preferable for different environments. As an example, the standard version of Skype uses a 'Wideband Codec' which provides very high quality sound, but needs a broadband IP connection, whereas codes such as AMR are more appropriate for networks where bandwidth is scarce (such as cellular networks).

2.0 CHALLENGES FACED IN VOIP DEPLOYMENTS

In this section, we describe four of the most common challenges that are faced in large scale VoIP deployments

2.1 NAT and Firewall traversal

From a deployment perspective, most users are behind a NAT and/or Firewall of sorts. Simply put, this provides unique challenges of 'reachability' for users behind such boxes. As an example, several enterprise networks offer 'private IP addresses' to computers. What this means is that the IP addresses are only understood within the domain they are allocated and are not globally reachable. Typically, in such a scenario, the NAT box converts from private to public IP addresses to and fro. NATing is a common solution to avoid the addressing space problems of IPv4 systems, where there may not be sufficient IP addresses allocated for all computers. Another example is broadband connections at our homes. We usually buy a DSL/Cable connection from our provider and pay for a single static IP address. However, most of us need to connect more than one PC to this connection. A common solution is to allocate private IP addresses to each internal PC and have the NAT box de/multiplex the connections using a combination of IP+port for the internal PCs. Technically, this is called NAPT (Network Address Port Translation) where IP+port is used to de-multiplex multiple simultaneous connections over a single public Internet IPv4 address. While all of this is great, NAT provides a unique challenge for protocols that embed IP addressing information as part of the packet payload. Consider, for example, a standard SIP message like so:

```
INVITE sip:bob@example.com SIP/2.0
From: sip:alice@example.com
To:bob@example.com
Contact:sip:10.4.16.12@example.com
.....
(rest of details)
```

In this case, 10.4.16.12 happens to be a private IP Address for the client in question. When this is inserted in an IP packet with source IP of 10.4.16.12, the NAT box smartly knows that 10.4.16.12 needs to be rewritten by a public IP+port so it can be routed back to the right client on receiving a response to this request. However, a traditional NAT does not know the contents of the payload, and hence, it does not change the IP address inside the SIP message.

In essence, this results in a situation where the response never reaches the originator. This is a classic example for the need of Application aware gateways which in addition to IP NATing are also capable of Application Level NATing. One possible solution is to deploy a NAT ALG at the network which in addition to IP level NATing also performs SIP NATing (These are called Application Layer Gateways - ALG). The problem with this solution is that it defeats end-end encryption. If the party behind the NAT wished to setup a secure signaling path with the called party, it would be mandatory for the ALG to obtain the encryption keys to be able to decode the payload for ALG operations. Therefore, the most desirable solution is for the endpoint to 'discover' its public IP address and insert the correct IP address in the SIP/SDP Payload. IETF has proposed several solutions to NAT traversal. The most common and simple solution of them all is a protocol specification called STUN. In essence STUN is a client/server protocol that requires a "STUN client" to be installed with the UA behind the NAT and a STUN server to be available on the public Internet which can respond to the client's requests by providing its "view" of the receive IP and port. STUN works for both TCP and UDP and it is expected that before a UA makes a call, it executes STUN procedures to discover its public IP address + port and encodes a SIP message accordingly.

One of the drawbacks of STUN is that it does not work with symmetrical NAT configurations. A Symmetrical NAT configuration is where the NAT boxes create a mapping based on both the source

IP+port as well as the destination IP+port. Simply put, this means that the IP address+port that the UA discovered via STUN, from the STUN server will be different from the public IP+port combination when contacting another external UA.

To solve this issue, IETF proposed a TURN solution where a 'TURN server' can act as a relay for symmetric NATs. In other words, if a symmetric NAT is discovered, the IP address of the TURN server can safely be used as the public IP+port for the UA behind the NAT irrespective of which outside UA it is contacting. The TURN server, in turn ensures that this relaying is only allowed for the selected calling/called parties providing additional security against general abuse.

Finally, even though TURN works for all NAT configurations, since it is a middle-box relay architecture, STUN is more desirable, if possible. Therefore, there was a need to specify a 'framework of discovery' that would automatically try various options and select the best option of NAT traversal. Thus ICE was born. ICE (Interactive Connectivity Establishment) is a procedure description of how UAs can try various options like STUN, TURN and other means to discover the best traversal protocol for a given scenario. Unfortunately, there are very few deployments that use ICE because of its complexity. (The IETF draft has been considerably simplified since it first was authored, so we hope to see more deployments sooner.)

NATing is not the only problem plaguing this space. Several Firewalls deploy several security measures which include opening symmetrical ports which means if a client behind the NAT opens port 6789 for media streaming, the UA on the far side cannot transmit media to another port. In addition to this, several firewalls close connections if there is a lack of traffic in a tunnel opened between a UA inside a firewall and one outside. To avoid this problem, several Session Border Controllers (SBC) implement fake pings for the sake of keeping the connection open. Firewall and NAT traversal is a detailed topic in its own right and readers are urged to refer to the links provided below for detailed analysis. Suffice to say that one of the most critical requirements for VoIP phones is their ability to 'auto-configure' and 'auto-discover' so that users do not need to worry about these complexities. Skype is an example of a successful auto-discovery solution that makes NAT traversal very transparent. Columbia university researchers earlier [dissected](#) Skype's NAT traversal and found that Skype uses a protocol very similar to STUN and TURN.

2.2 Quality of Service (QoS)

Quality of Service is a much talked about problem space for Internet Telephony. The basic problem statement is "If the Internet is a massively distributed bunch of routers with no guarantee of which voice packet gets priority over another, how does one ensure that quality of service is consistent ?" There are different ways of looking at this problem. From one perspective, there are several reports from MCI which talk about the fact that the Internet core network backbone is 98% unutilized and QoS is lesser of an issue than vendors make it out to be. The other side of opinions state that instead of focussing on QoS priorities, optimized codec can be used to reduce bandwidth utilization. While these arguments are fine for the core backbone and for non-critical applications, QoS becomes a serious issue when one considers (Over The Air) transmission for VoIP. Radio resources are scarce, and in mission critical applications, consumers will want an SLA which states guaranteed Quality of Service.

There are several challenges to QoS - both from a implementation and a deployment perspective. With WiFi related technologies being deployed for the first hop access for VoIP phones, noise and interference become significant issues that can degrade overall user experience and proper QoS allocation schemes can help reduce these problems. QoS addresses several real-life problems with VoIP which include traffic shaping (high bandwidth applications which hog shared bandwidth can be controlled/limited), Avoiding Congestion (implement heuristic algorithms such as Weighted Random Early Detection (WRED) (where the packet Queue is dynamically adjusted by analyzing traffic priority and allocating more space for higher priority packets) and similar. Proper QoS management is critical for deployment of VoIP in large scale networks where first hop bandwidth is restricted, such as modern deployments of 3GPP cellular networks.

2.3 Trust models and Identity

One of the biggest challenges of using the Internet as a medium for communication is that any mode of communication is inherently open to any security concerns of the way the Internet is deployed today. One of the most common issues is that of Identity and Trust. In the context of the discussion here, we are talking about subscriber and network identity and trust and not lower level security issues such as IP Spoofing. IP Spoofing is a concept where a rogue client can use a raw socket interface and spoof it's IP to belong to other nodes. There are several ways to address IP level spoofing including strong sequencing [algorithms](#) packet filtering etc.)

At the application level, the basic questions are:

"Can the Network trust the subscriber ?"

"Can the subscriber trust the network ?"

Since the Internet is, as described earlier, a distributed environment with no central control, both of the questions above are important to assess. As an example, a rogue client may easily send a "From:bob@yahoo.com" in a SIP INVITE message but the network needs to assess if the advertised identity is really the correct identity. Common mechanisms include authentication of users using algorithms such as Digest, IMS-AKA and other secure mechanisms. Yet another concept that has gained acceptance is "Network Asserted Identity". In this scheme, even if the user has been authenticated, the network has the right to override or supplement this identity with a "Network Asserted Identity (NAI)" which is used by other nodes in the network. NAI as a concept is very useful to apply additional security policies (example, Bob has authenticated correctly, but he is using an IP domain that was never used before - so let me add an NAI which can be an indication to network services to possibly de-activate certain sensitive services provisioned to Bob). The second question is also equally important - can the subscriber trust the network ? To address this issue, deployments typically use a mutual authentication mechanism where both the sides can verify each other. One example is establishing an IPSec tunnel between the client and the network.

2.4 Spam over Internet Telephony

There has been a lot of discussion in the past on how serious of a problem spam for Internet Telephony (SPIT) and Spam for Instant Messaging (SPIM) is as VoIP deployments increase their market share.

Spam itself as we all know is all-pervasive in the email world. I was reading an interesting [report](#) from Symantec which reports that 67% of email is spam these days. While the percentage is staggering, there is some(?) comfort in knowing that this percentage has stabilized, which might mean that spam filters/gateways are maturing at a rate that is able to cope with the mutation of spam tricks.

The interesting thing is that even though pundits scream about the problem of spam over Internet Telephony, not too many carriers are biting, at least, for now. It's not that they don't think its important, but it just seems that they have other problems to solve (like it or not, security is the hardest and the last solved problem in real life).

2.4.1 Is SPIT a real problem ?

In short, SPIT is a problem, that will eventually surface. The logic: the infrastructure needed for SPIT is very similar to the infrastructure needed for spam. The cost is very low, and there are no [Do-not-call](#) registries on the internet (at least for now). In addition, it is harder to detect a valid IP address of source as compared to an originating phone number. All in all, the Internet offers SPITers better anonymity and lower costs – so why not ? Infact, there are [companies](#) who already think it is a big problem and have products out. Not to mention that this space will soon be chock-a-block with [patents](#).

However, I really don't think SPIT is a huge problem for "walled-garden" networks - it's a much bigger problem when you have peer2peer networks.

2.4.2 Why is SPIT a bigger challenge than SPAM ?

There are a few reasons for this:

- **More Intrusive:** a telephone is intrusive by nature where as email is not: when you get a phone call, it rings, and loud. You cannot just ignore the ring, nor can you 'move on to the next call' – you need to answer it. Compare that to a spam mail, which you can choose to ignore and move ahead and eventually move to junk
- **Harder to detect:** Today's spam filters are fairly advanced. In addition to white and black lists, a lot of them implement heuristics which try and detect the language patten to filter out possible spam guised as valid content. With SPIT, you are not looking at text -its a voice. Detecting heuristics in a voice stream is much harder - add to that the fact that a single sentence can sound very different depending on who is speaking (accent)

2.4.3 SPIT in Walled Garden vs. Peer2Peer networks

One of the strongest and most effective ways to avoid SPIT is by good authentication. In an ideal world, if every user could be authenticated for its identity, the chances that a spam bot gets to you reduces greatly. Walled garden networks usually operate in "circles of trust". Lets suppose bob@verizon.com were to call sue@att.net via SIP. For the att.net proxy to accept the call from bob, it would most likely expect a digital authentication certificate from Verizon's proxy telling the AT&T proxy that this call actually came from verizon. In other words, AT&T trusts the Verizon network and expects the Verizon network to have authenticated its users. This sort of 'trust circles' between service providers is common. AT&T cannot go ahead and authenticate each and every user on a foreign domain, so instead it decides to trust the network, based on strong authentication. It is verizon's responsibility to make sure that bob is a valid user. For a spammer to break such a network would require that

- a) It manages to successfully authenticate with verizon (which may well be an identity theft case) ,or,
- b) Manage to hijack the network authentication certificate and key to be able to 'fake its network identity' (which is not easy to do), or
- c) Be able to reach sue's phone directly, bypassing both the proxies.

Case c) works if the UA itself accepts calls from any source. A secure UA should be configured to reject any calls that are not passed on to them from their inbound proxy via TLS. In other words, if bob@verizon.com were to call sue@att.net, Sue's UA should detect if it came along with credentials from its authorized inbound SIP proxy and if not, it should be rejected, prompting the caller to go through the proxy. If all UAs were indeed configured this way, then the problem of SPIT reduces a great deal (not eliminated, but significantly reduced). However, not all UAs are configured this way. In addition, while this can be enforced within a walled garden network, problems arise when:

- The UA's are part of a peer2peer network where there may not be any central agreements of trust
- An adhoc-network tries to call into a walled garden network (Say, your uncle in Korea has set up his own [FWD](mailto:fwd.pulver.com) SIP phone and tried to call into you, a member of MCI's SIP network and MCI has no idea about [@fwd.pulver.com](mailto:fwd.pulver.com) - just an example)

One way around it is for UA's to allow caller authentication. In other words, even if a call comes via unknown channels, instead of rejecting it, challenge the caller for identity, while at the same time, don't make it cumbersome for the called user (for example, if every SPIT phone rang your phone, and then challenged it, you'd switch back to PSTN within a day ! At least in PSTN, the FCC has done an effective

job with the DoNotCall registry). For example, Vonage, which is an example of a Walled Garden implementation requires authentication at the very least to let calls in.

2.4.4 Various ways to address SPIT

SPIT, just like its older sibling spam, is a prime example of "many partial solutions lend to a stronger overall solution". In other words, there is no one mechanism that can effectively eliminate SPIT. The solution is to deploy various levels of defence in the hope that one of them catches the call before it reaches you. Here are some of them (since we mentioned, that content filtering which works with spam is mostly useless with voice)

- Strong Authentication- is an important first step in filtering SPIT. As discussed in detail above, one of the best ways to filter SPIT is by network and UA participation, where the UA accepts calls only from a TLS route from its inbound proxy and the Network authenticates users. However, this is far from the current deployment situation (many UAs don't yet support TLS).
- Reputation Based Systems- in addition to network identity, a reputation system works by assigning scores to sources. This score is a statistical formula based on its history. As an example, if a spitter "seller@tek.ru" makes a lot of calls to a network, and the users 'flag' the call as a "bad call", then, depending on several factors, that identity could be marked as 'bad' and this reputation could be distributed across the network to warn other users. There are several challenges to this:
 - Spam agents often change identities
 - It is easy to poison such a system - for example, force negative feedback even when it's not true. In other words, a good reputation system is a harder solution than it sounds
- Central black lists- Not a complete solution, but an effective one. Spammers will keep creating new addresses while the black lists will keep adding to their repository. The lists eventually get more complete and more effective. Such [lists](#) exist today, and are very helpful.
- Puzzles - Another mechanism is when a call arrives from an unknown source, throw an automated challenge. An example: If I receive a call for the very first time from user "Joe" instead of ringing my phone, redirect him to an IVR that asks "Joe" to press some random 4 digit sequence as a security verification. This is an irritant for Joe, especially if he is a valid user – but needs to be done only once. For this to work, however, either the called UA or the called UA's network server needs to remember whether this is a first call, or whether Joe has called before. I personally think this is an effective solution, albeit with an irritant factor for the caller, but hopefully only once. The bigger concern, however, is who remembers this is the first call or the 1 millionth, especially when thousands call you over years.
- Payment systems - Companies like Microsoft pushed this hard for emails. The idea is simple - for you to make a call, you first need to deposit a payment via some payment gateway for the called network. If the call is accepted, you get refunded, if not, you lose your money.

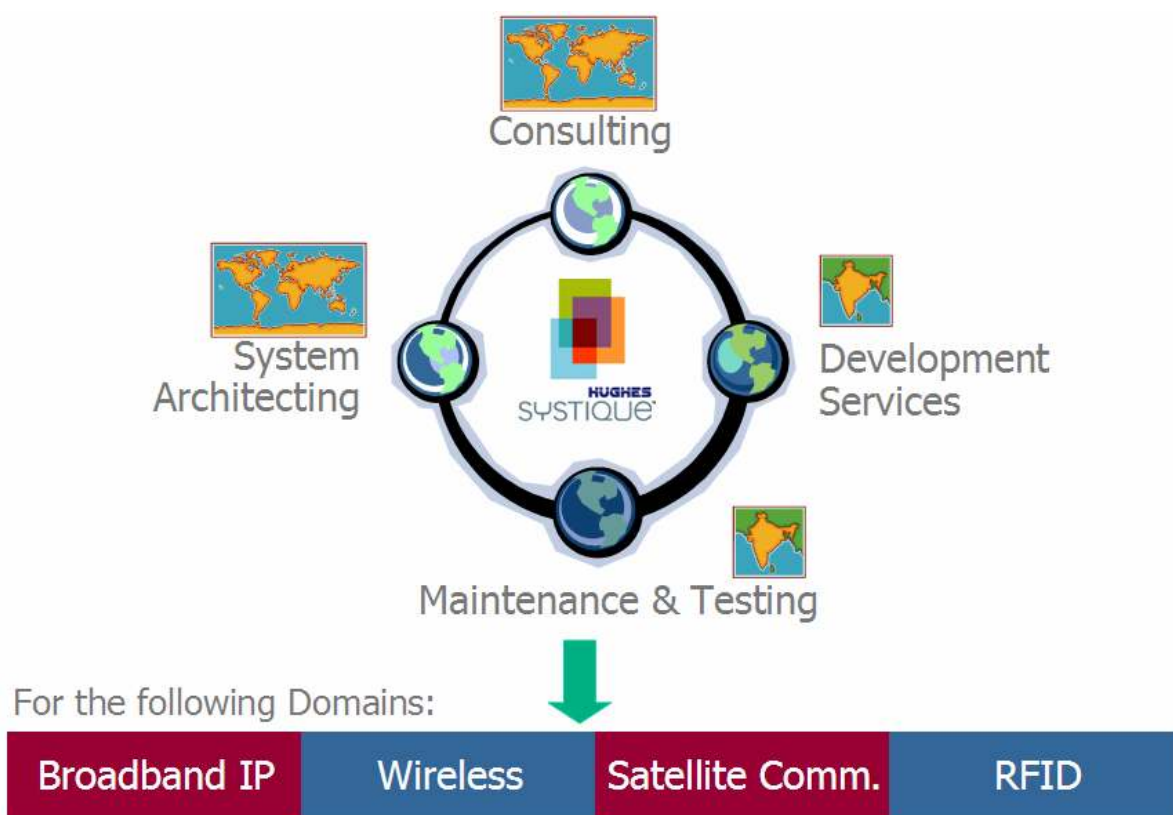
3.0 REFERENCES

<http://asrg.sp.am/>
<http://www.ietf.org/internet-drafts/draft-schwartz-sipping-spit-saml-00.txt>
<http://www.ietf.org/internet-drafts/draft-ietf-sipping-spam-01.txt>
<http://www.jdrosen.net/papers/draft-rosenberg-midcom-turn-08.txt>
<http://www.jdrosen.net/papers/draft-ietf-mmusic-ice-08.txt>
<http://www.iptel.org/ietf/firewall/nat/rfc3489.txt>

APPENDIX A ABOUT HUGHES SYSTIQUE CORPORATION

HUGHES Systique Corporation, part of the HUGHES group of companies, is a leading communications Consulting and Software company. We provide Consulting, Systems Architecture, and Software Engineering services to complement our client's in-house capabilities. Our "Best Shore" model coupled with an experienced management and technical team team is capable of delivering a total solution to our clients, from development to deployment of complex systems, thus reducing time, risk and cost

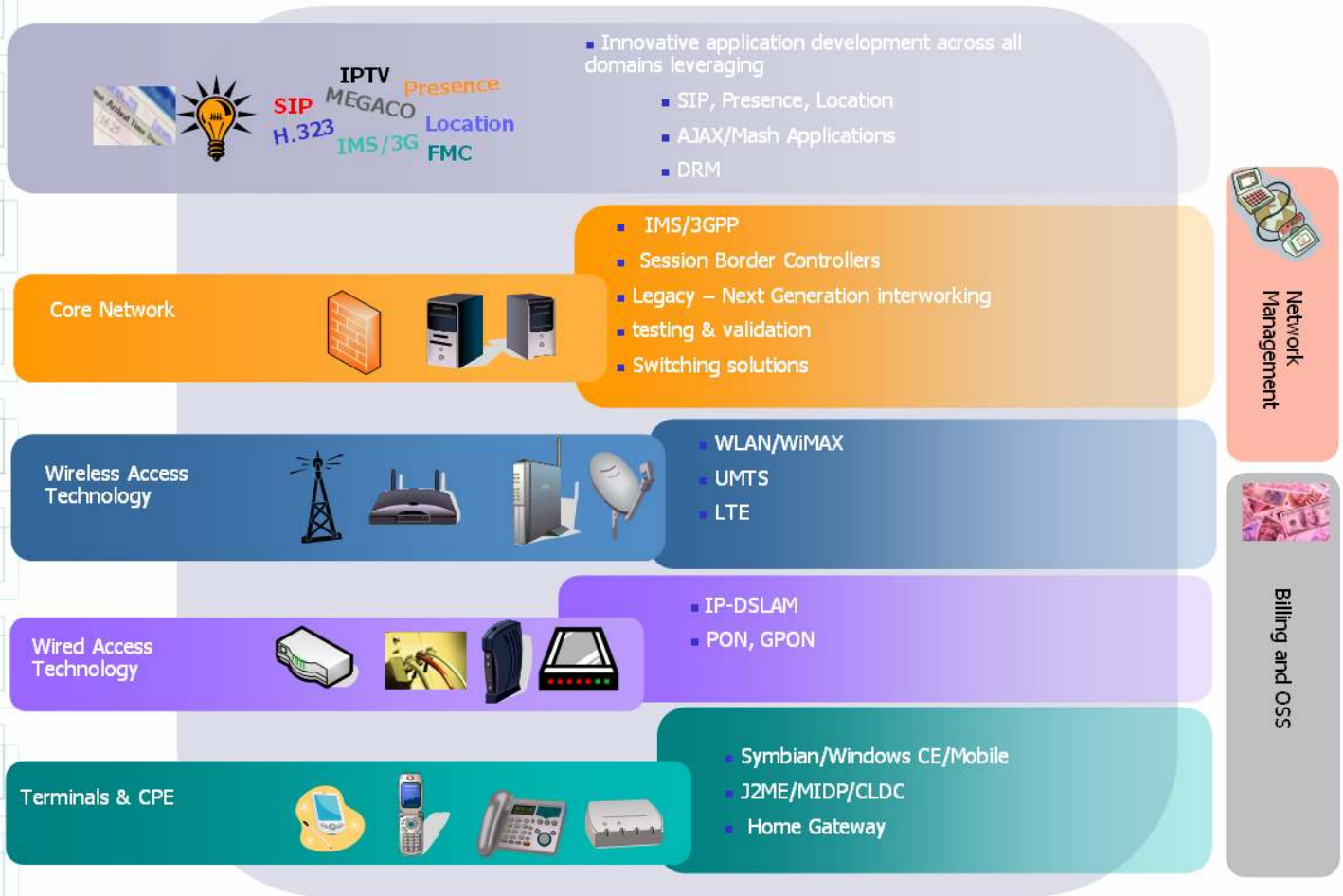
HSC Solution Space:



CONTACT INFORMATION:

phone: +1.301.527.1629
 fax: +1.301.527.1690
 email: whitepaper@hsc.com
 web: www.hsc.com

HSC Expertise Areas in Brief:



CONTACT INFORMATION:

phone: +1.301.527.1629
 fax: +1.301.527.1690
 email: whitepaper@hsc.com
 web: www.hsc.com