

SIRIUS

Custom Router Manager

Interconexión API con Node.JS

Autor: Andrés Genovez Tobar

SPM SOLUCIONES TICS

CUENCA -ECUADOR

MTCNA / MTCTCE / MTCRE / MTCWE

APPLE ACMT

ELASTIX ECE

LINUX LPI-1 / UBUNTU UCP

EMAIL: agenovez@cspmsa.com

Web: <http://www.crice.org>

Objetivo

- Tomar control sobre los clientes en un solo programa.
- Permitir la administración de Mikrotik por personal sin conocimientos técnicos.
- Tener un registro completo de accesos y cambios.

Que es API

- Interfaz de Aplicación Programable
- Interconexión entre Aplicación Desarrollada en cualquier lenguaje y Mikrotik
- API viene deshabilitado por defecto, desde la versión 6.1 de ROS viene habilitado por defecto e incluye API sobre TLS (SSL)
- [admin@MikroTik] > ip service enable api

Diagrama API



SIRIUS

- SIRIUS Utiliza Mikronode Copyright Brandon Myers.
- Creado usando Node.JS
- Crear un sistema completo (Software Libre)
- Acceso y registros a base de datos.(MongoDB)
- Seguridad
- Aplica las bondades del Software Libre
 - Código fuente accesible
 - Control de Errores
- <http://www.github.com/siriuscrm>

Conceptos

- Un programa desarrollado con un conector API NO es un reemplazo de Winbox.
- Que es Node.JS?
- Porque utilizarlo en esta aplicación?
 - Limites APACHE
 - Web 2.0
- Porque Software Libre?
 - Seguridad
 - Escalabilidad
 - Multiplataforma
 - Tienes el control
- Aprender y compartir.

Diagrama de Sirius

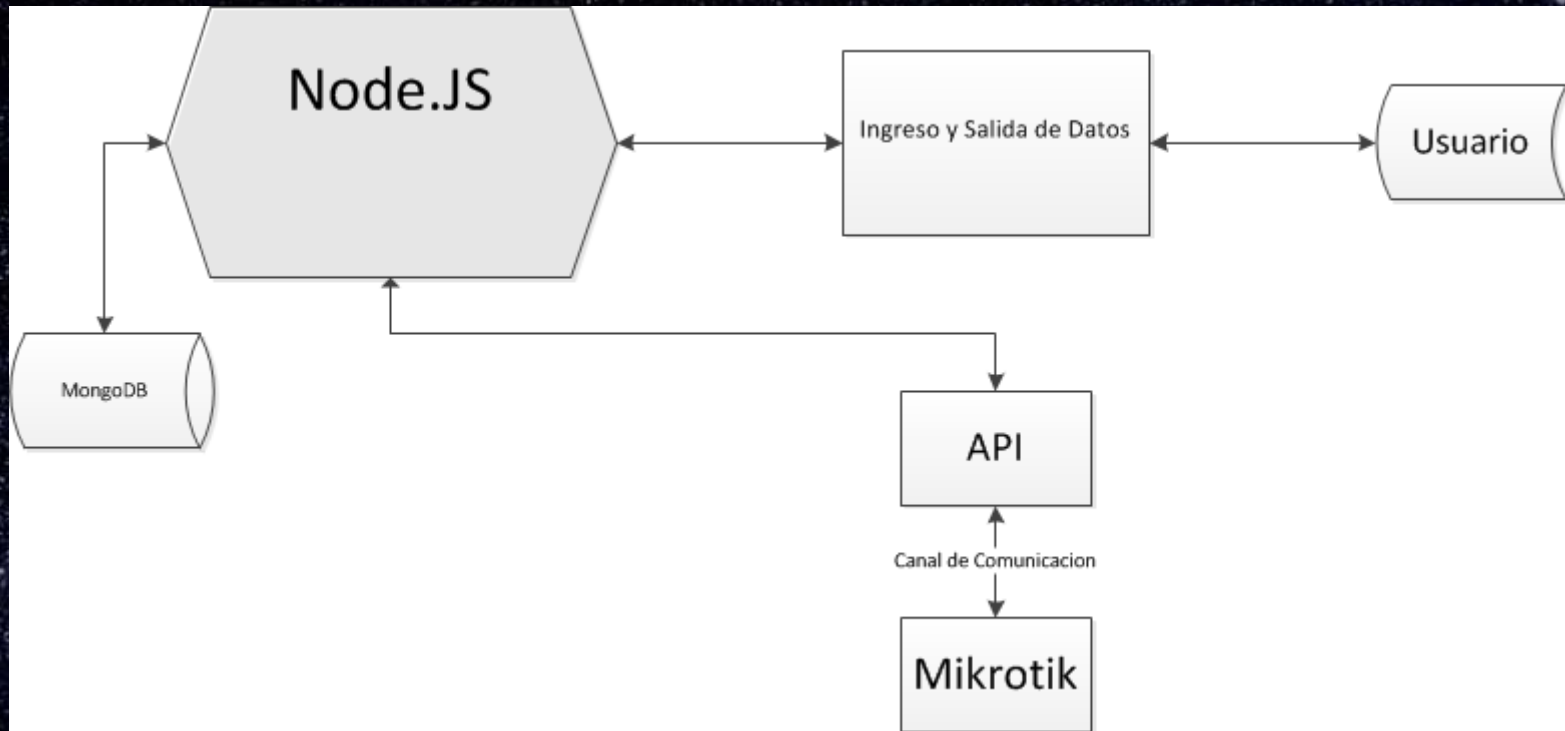
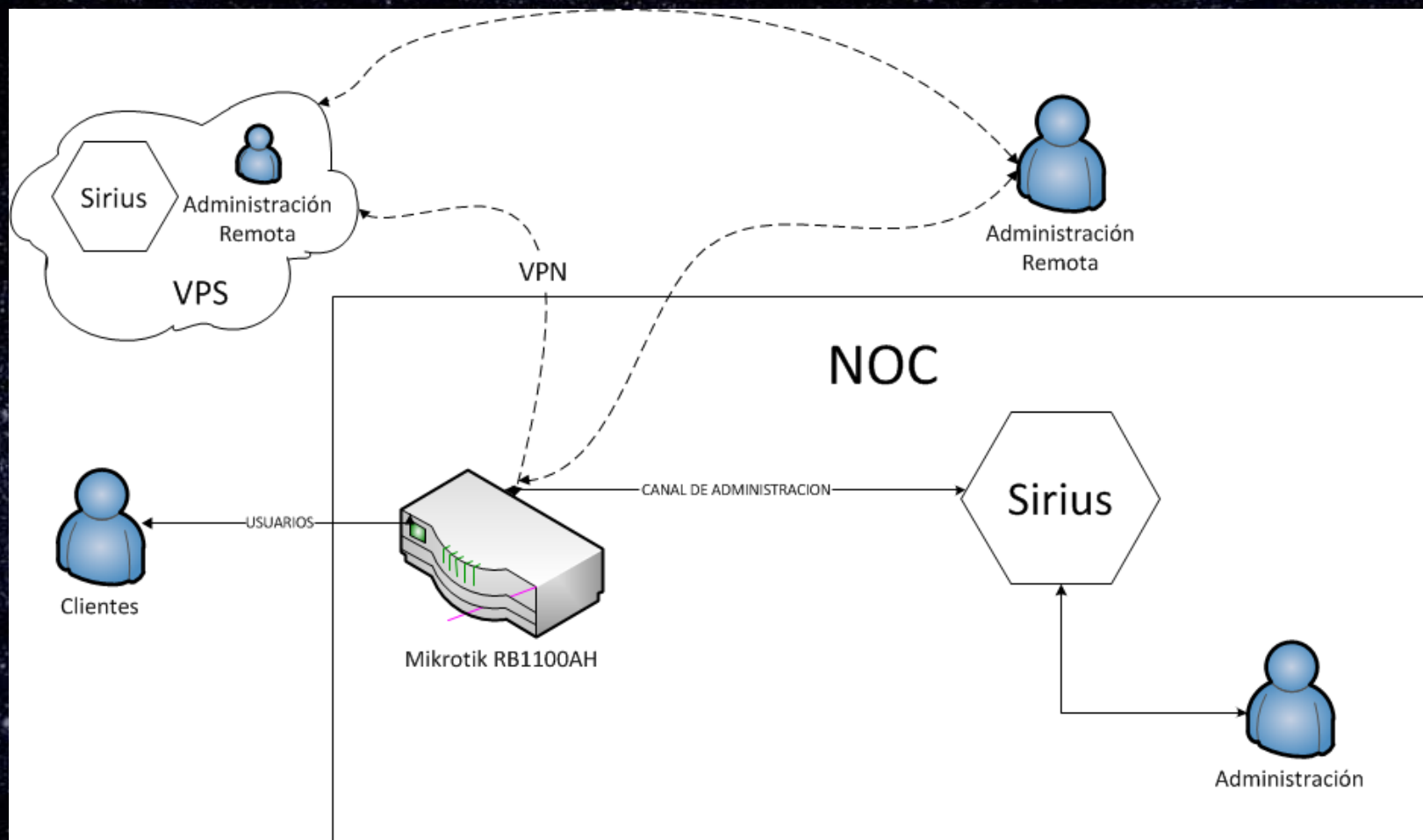


Diagrama Sirius Implementado



Público Objetivo

- Administradores / Help Desk
 - Sin conocimiento Técnico
 - Tareas específicas
- Desarrolladores
 - Conocimientos de programación (objetos)
 - Compartir conocimiento
- Cualquiera

**ERA LA CASA DE UN CLIENTE
MOROSO...**



ERA...

Ejemplo de llamada

- `var api = require('mikronode');`
- `var connection = new api('192.168.33.1', 'admin', 'clave123');`
- `connection.connect(function(conn) {`
- `var chan=conn.openChannel();`
- `chan.write('/ip/firewall/address-list/getall',function() {`
- `chan.on('done',function(data) {`
- `var parsed = api.parseItems(data);`
- `parsed.forEach(function(item) {`
- `console.log("Session: %j", item);`
- `});`
- `chan.close();`
- `conn.close();`
- `});`
- `});`
- `});`

Resultados de la Llamada

- Session: {".id": "*314", "list": "impago", "address": "172.16.26.2", "dynamic": "false", "disabled": "false"}
- Session: {".id": "*315", "list": "impago", "address": "172.16.26.3", "dynamic": "false", "disabled": "false"}
- Session: {".id": "*316", "list": "impago", "address": "172.16.26.4", "dynamic": "false", "disabled": "false"}
- Session: {".id": "*317", "list": "impago", "address": "172.16.26.5", "dynamic": "false", "disabled": "false"}
- Session: {".id": "*318", "list": "impago", "address": "172.16.26.6", "dynamic": "false", "disabled": "false"}
- Session: {".id": "*319", "list": "impago", "address": "172.16.26.7", "dynamic": "false", "disabled": "false"}
- Session: {".id": "*31A", "list": "impago", "address": "172.16.26.8", "dynamic": "false", "disabled": "false"}
- Session: {".id": "*31B", "list": "impago", "address": "172.16.26.10", "dynamic": "false", "disabled": "false"}
- Session: {".id": "*31C", "list": "impago", "address": "172.16.26.12", "dynamic": "false", "disabled": "false"}

Multiples Llamadas

- Multiples llamadas requieren un canal diferente, abrir y cerrar cada canal.
- Ejemplo: Subida CSV
- Importante.

Ejemplo Practico

Conclusiones

- Subir 160 clientes tomar un tiempo considerable. (csv) Varias horas, días, semanas.
- Ingreso manual propenso a errores.
- Dar acceso aún con restricciones puede producir un error ejemplo (impago = 0.0.0.0)
- Se pueden alcanzar las necesidades o tareas repetitivas o específicas inalcanzables.

Futuras Versiones

- Completar el desarrollo.
- Integración con base de Datos
- Mejorar Código
- Mejorar Seguridad
- Útil
- Modular
- Código Fuente Abierto.

Preguntas

Muchas Gracias

Andrés Genovez Tobar
SPM SOLUCIONES TICS

Twitter: bitfrost