



```
y HOIDDICT = <OIDDICT> );
heck = "SiteNewsYupdateTimey";
```

-e sti pent B 11 (SC SI1)

e156

heck .

ch my SOID (BOIDDICT)

"Information is a source of learning. But unless it is organized, processed, and available to the right people in a format for decision making, it is a burden, not a benefit."

-William G. Pollard

Information Goals

- Provide as much data to the public as possible
 - CCTV images

neck

-e Sri

- RWIS atmospheric/roadway data
- HAR messages
- CMS messages
- Accurate, Timely, and Reliable
 - Ensures that users trust the information
- Clearly formatted and well documented

y HOIDDICT = COIDDICT>); heok = "SiteNamb9UpinteTimey";

header based Background

heck . (-e sti pent B

Stmp

ch my SOID (BOIDDict)

 \mathbf{O}

 All TCP/IP based element network Large Dial-on-demand network

– POTS (32 Sites)

- ISDN (26 Sites)
- Microwave Radio network
 - 3 Backbone links
 - 10 Roadside links

DICT, y HOIDDICT = <OIDDICT>); Network Segregation heok = "SiteMashyUpdate ch my SOID (BOIDDict) (@tmp = split("1",SOID); neck .= "\$cmp[0] 9"; SErrorLog); Administrative Network Field Element Network (FEN) Controlling field devices Email - CCTV (PTZ) Web browsing - Changeable Message **Document retrieval** Signs (CMS) Word Documents Retrieving data from field 0 Images devices

"", "\$cauOutput") of die "Cannot Open

(flock(RWISDATA, LOCK_EX(LOCK_NB)) (

if (\$debug) (print my stimet = 0;

open(RWISDATA,

- CCTV (Video)
- Roadway Weather Information Systems (RWIS) data



- Push
 - Process device initiates the communication to external server to "push" or "deposit" information



open(RWISDATE ANTA LIFLOCKI RUISDATA,





open(BWISDATE, Exporter is waiting to

```
W BOIDDICT = COIDDICT> );
heck = "SiteNambyOpdateTimey";
```

ch my SOID (BOIDDict)

Stmp heck .

wortte header based Deficiencies

Dial on demand limitations

- Only so many modems available for dialing to remote sites
- Unknown number of external users on FEN
 - Possible overloading
 - Vulnerabilities
- Testability
 - Too many critical processes are not within our control to troubleshoot and repair
 - Cannot test and troubleshoot at every "hand-off" to verify working condition

Deficiencies (cont)

Reliability

g BOIDDict = (OIDDICT>);

ch my SOID (BOIDDict

Otmy . neck .

- Too many "hand-offs" between devices
- Too many system administrators working on servers outside of our control without proper knowledge of all processes
- System administrators can accidently kill critical processes (without realizing it) while working on servers

Oversight

 Good system administrators should know ramifications of working on server, but this is not so in current environment

Need for Change

Simplify the process

g BOIDDict = (OIDDICT>);

ch my SOID (BOIDDict

Stmp heck .

(-e SE

- Cut out unneeded devices and processes in grabbing information
- Limit the amount of "hands in the pool"
- Limit field element network traffic originating from administrative network

	A A A A A A A A A A A A A A A A A A A		
heck = "Sit ch my \$OID gtmp = = heck	(EOIDDict) ((EOIDDict) ;		
-e \$ri pen(P	The sizes		
1050] 11 (50' 511)	I ne simpi	ler a process can]
ope) for P)	be, the le	ess there is to	
210)) else	break or	go wrong.	
pri pri			
toreach giork my ipid	= forever.	xpdffTlmm [#] } == l. P/	
11(1) 1f (\$d 1f (\$d 1f (\$	porcing file child process	10×11/10 ⁻¹²)	
# 11 my opi	(Sdebug) (print "Exportance Stimer = 0; en(RWISDATA, "c", "StauOutput") of (en(RWISDATA, LOCK_EX)LOCK_N () (flock(RWISDATA, LOCK_EX)LOCK_N	die "Cannot Open Wille. HB)) (xporter is wälting in rend file. (timet)n";)	

DDICT, "<", \$OIDDIGT) of die logifi(un IDDICT, LOCK_SH);

g BOIDDICT = <OIDDICT>); DDICT;

14

```
y HOIDDICT = <OIDDICT> );
heok = "SiteNambyUpdateTimey";
```

ch my SOID (BOIDDict)

Stmp heck .

(-e sti

Solution

Use reliable single process device \bigcirc

- Use device to pull data from the field/local devices
- Use device to push information to external sources
- Use engineering approach to develop process and modules



```
y BOIDDICT = COIDDICT> );
heok = "SiteNambyUpdateTimey";
```

ch my SOID (BOIDDict)

Atmp heck .

-e sti

Hardware

Moxa DA-661 Specifications

- Computer
 - Intel XScale 533 MHz processor
 - 128 MB DRAM onboard
 - 32 MB Flash onboard
 - PCMCIA cardbus

```
W HOIDDICK = COIDDICT> );
heok = "SiteNambyUpdateTimey";
```

Hardware

Moxa DA-661 Specifications (Cont)

- I/O

ch my SOID (BOIDDict)

Stmp heck .

-e SE

- 2 LAN 10/100Mbps (RJ-45)
- 16 RS-232/422/485 8-pin (RJ-45) (software config)
- Operating System
 - Linux 2.6.10
 - File system: JFFS2
 - Internet protocols
 - TCP, UDP, IPv4, SNMPv1, ICMP, ARP, HTTP, CHAP, PAP, SSH 1.0/2.0, SSL, DHCP, NTP, NFS, Telnet, FTP, TFTP, PPP, PPPoE

```
y BOIDDICT = COIDDICT> );
heok = "SiteNambyUpdateTimey";
```

ch my SOID (BOIDDICT)

Atmp heck .

-e sti

Hardware

Moxa DA-661 Specifications (Cont)

- Operating System (cont)
 - Apache
 - Terminal Server (SSH)
 - Includes Perl 5.6.2
 - Working with Moxa for newer version

```
W BOIDDICT = COIDDICT> );
heok = "SiteNambyUpdateTimey";
```

ch my SOID (BOIDDICT)

Stmp heck .

Moxa Pros

- Relatively cheap embedded device
- Easy to set up
 - Technician level employee can install
- Everything can be stored on removable flash card
- Failure can be fixed/restored by technician level employee
- Failure of device only affects a single data set of information
- Two LAN ports allows device to be on two separate networks
- Rack mountable
- Environmentally Hardened
- Perl 5.6.2 is native
 - Working with Moxa for newer version



ch my SOID (BOIDDict)

Atmp heck .

-e SE

Moxa Cons

- Clock Drift

- Up to 45 seconds per hour
- Possible Kernel bug
- Working wit Moxa to address problem
- XScale Processor
 - Executables binaries require cross compilation



Push

 Process device initiates the communication to external server to "push" or "deposit" information

Engineering Approach

Develop reliable program

- Is able to catch and log all possible errors
- Is able to recover from and log fatal errors
- Modular design

- COIDDICT>):

heak

(-e sri

- Effective for troubleshooting
- Well defined inputs and outputs
- Checks if inputs are existing and are within reason
- Checks if outputs are within reason before returning, or returns error status









```
W BOIDDICT = COIDDICT> );
heok = "SiteNamb@UpinteTimey";
```

Scheduler

ch my SOID (BOIDDict) **ITS Node**

heck .

0

- Roadside LAN serves as 'Access point' to field element network
- Aggregation point for multiple field elements (CCTV, RWIS, CMS, ect)
- Web user interface for adding new or modifying existing ITS Nodes
- Defines schedule for each ITS Node based on algorithm to account for communication type and number of existing out going lines.
- Pushes 'Timing' file to scheduled devices
 - RWIS Information Relay
 - CCTV Image Relay
- Runs NTP server for scheduled device
 - Keeps clocks between multiple devices synchronized

Exporter is vaiting

```
DICT,
IDDICT, LOCK_SH ):
y HOIDDict = <OIDDICT> );
heok = "SiteNambyUpdateTimey";
```

ch my SOID (BOIDDict)

Otmp

heck .

pent B

Scheduler

Timing file example

Site Name Anderson Grade; Antlers Smt; Black Butte; Bogard; Buckhorn; Doyle; Dunsmuir; Fredonyer Smt; Fredonyer East; Hatchet Mtn; Hilt; Hornbrook; Janesville; North Weed;

IP Address	Com	Minute
10.xx.xx.xx;	ISDN;	0,37
10.xx.xx.xx;	POTS;	5,42
10.xx.xx.xx;	ISDN;	10,47
10.xx.xx.xx;	POTS;	15,52
10.xx.xx.xx;	ISDN;	20,57
10.xx.xx.xx;	POTS;	25,3
10.xx.xx.xx;	ISDN;	30,8
10.xx.xx.xx;	POTS;	35,13
10.xx.xx.xx;	POTS;	40,18
10.xx.xx.xx;	POTS;	45,23
10.xx.xx.xx;	POTS;	50,28
10.xx.xx.xx;	POTS;	55,33
10.xx.xx.xx;	POTS;	2,38
10.xx.xx.xx;	POTS;	7,43

open(RWISDATE AND DETOCKI RUISDATA,

Exporter is valting to





UNITE LIFTOCKI RUISDATA, LOCK_EXILOCK_N Exporter is valting to read file open(BWISDATL,





CCTV Image Relay

Process Hash

y HOIDDICT = <OIDDICT>);

heck = "SiteNamt(UpA ch my \$0ID (BOIDDict)

Otmp

heck .

-e SE

pent R

- Site Name, IP Address
• Only for sites to grab image this minute
Black Butte => 10.xx.xx.xx
Doyle => 10.xx.xx.xx
Dunsmuir => 10.xx.xx.xx

Hilt => 10.xx.xx.xx I5-SR44 => 10.xx.xx.xx

CCTV Image Relay

comType Hash – Site Name, Communication type

y HOIDDICT = <OIDDICT>);

heck = "SiteNamt(Upd ch my \$010 (BOIDDict)

Stmp =

heck .

-e Sri

pen(R

- POTS, ISDN, MW
- Used to set timeout value for connecting

Black Butte => ISDN

Doyle => POTS

Dunsmuir => ISDN

Hilt => ISDN

I 5 - S R 4 4 => MW



open(RWISDATE, RWISDATE, LOCK_EXTENSION (flock(RWISDATE, LOCK_EXTENSION) to reast



open(RWISDATA, LOCK_EXTRACTOR IS USIDING to really (flock (RWISDATA, LOCK_EXTRACT) ; superver is usiding to really the second second



TOUR

38



Buckhorn CCTV/RWIS

heck = "SiteName(Upin sch my SOID (BOIDDict) @tmp = mp heck = "1

(-e stanout

pen(RWISD homp(my (slose RWISD if (Scheck Sfile(0)

> open(R) foreach print

close

open(RW print RW close R

foreach m stork h my Spid

3 11 (# #1 1





W BOIDDICT = COIDDICT>); **RWIS Information Relay**

Process Hash

neck

Stmp heck .

(-e Sri

- Site Name, IP Address
 - Only for sites to be interrogated this minute exportTime, 1 is used to export this minute Black Butte => 10.xx.xx.xx
 - Doyle => 10.xx.xx.xx
 - Dunsmuir = > 10.xx.xx.xx
 - => 10.xx.xx.xx Hilt
 - Hatchet Mtn => 10.xx.xx.xx

y HOIDDICT = <OIDDICT>); **RWIS Information Relay**

comType Hash

neox

heck .

-e sri

- Site Name, Communication type
 - POTS, ISDN, MW
 - Used to set timeout value for connecting
 - Black Butte => ISDN
 - Doyle = > POTS
 - Dunsmuir = > ISDN
 - Hilt = ISDN
 - Hatchet Mtn => POTS





open(BWISDAIR, RWISDATA, LOCK_EXTRACTOR is value to fee







open(RWISDATA, LOCK_EATDONCE (flock(RWISDATA, LOCK_EATDONCE) is waiting to fem



open(RWISDATA, LOCK_EATHONIC (flock(RWISDATA, LOCK_EATHONIC) (flock(RWISDATA, LOCK_EATHONIC) (flock(RWISDATA, LOCK_EATHONIC) (flock(RWISDATA, LOCK_EATHONIC)



open(RWISDATE, RWISDATE, LOCK

Exporter is valuing to



open(RWISDATE ALLA LIFLOCKI RUISDATA,



object, "<", \$01001et) or die logarity " IDDICT, LOCK_SH); y BOIDDICT = COIDDICT>); Peox = "Streppendloburglined." ch my SOID (BOIDDict) (Stmp = split("1", SOID): neck .= "\$ump[0]}"; (-e \$rawOutput) (pen(RWISDATA, "<", "StavOutput") homp(.my gfile = <RWISDATA>); CLOSE RWISDATA: if (\$check ne \$file[0]) (\$file[0] = \$check; open(RWISDATA, "+>", "\$rawOutput foreach my \$row (Bfile) (DEADE RWISDATA STOR . "AB"S close RWISDATA:) else (open(RWISDATA, "Do", "Schwourpu print RWISDATA Scheck . "\n"; CLOSE RWISDATA: foreach my skey (sort keys sprocess if (\$debug) (print "Child for if ((skey eq "expectition").46 if (\$debug) (print "Export. my Stimer = 0; open(RWIEDATE, "c", "Scau DELLA DELOCKI RUISDATA, 1



: ((\n", SErrorLog))





open(BUISDAIA, BUISDATA, LOCK_EATDONC







open(BWISDAIR, RUISDATA, LOCK_EATDONC

```
#Sets up correct header line
IDDICT, LOCK_SH );
y HOIDDICT = COIDDICT> );
                                                                          peader based
                                       Questions?
Peox = "Streppendloburglined."
ch my $0ID (BOIDDict) (
@tmp = split("1", SOID):
heck .= "$cmp[0] ";
pen( RWISDATA, "d", "SrawDurput" ) or die logErr("EPROR : PGISDATI : Si'n", SErrorLog);
                                                            Sopen file to zero out and write
homp( my gfile = <RWISDATA> ):
 Open( RWISDATA, "+>", "$rewOutput") or die logErr("ERROR : RWISDATA : 3/\n", $Errorbog);
                                                             floops through each tou to add \n
CLOSE RWISDATA:
if ($check ne $file[0]) (
$file[0] = $check;
  foreach my $row (Bfile) (
   PEANE RWISDATA SCOR . "An":
 Open( RWISDATA, "oo", "IchwOutput") or die logErr("EPROR : AWISDATA : Cannot greate file : $1\a", $ErrorLog);
) else (
 print RWISDATA Scheck . "Vn";
  close RWISDATA:
foreach my $key (sort keys sprocess) (
    if ($debug) ( print "Child for $key Started...\n"2)
    if (($key eq "experitive") is ($process("experitive") as 1 )) (
       open( RWISDATA, "<", "$rayOutput" ) or die "Cannot Open $1\n";
        STATE LIFTOREL RUISDATA, LOCK EX(LOCK_NB)) (
```