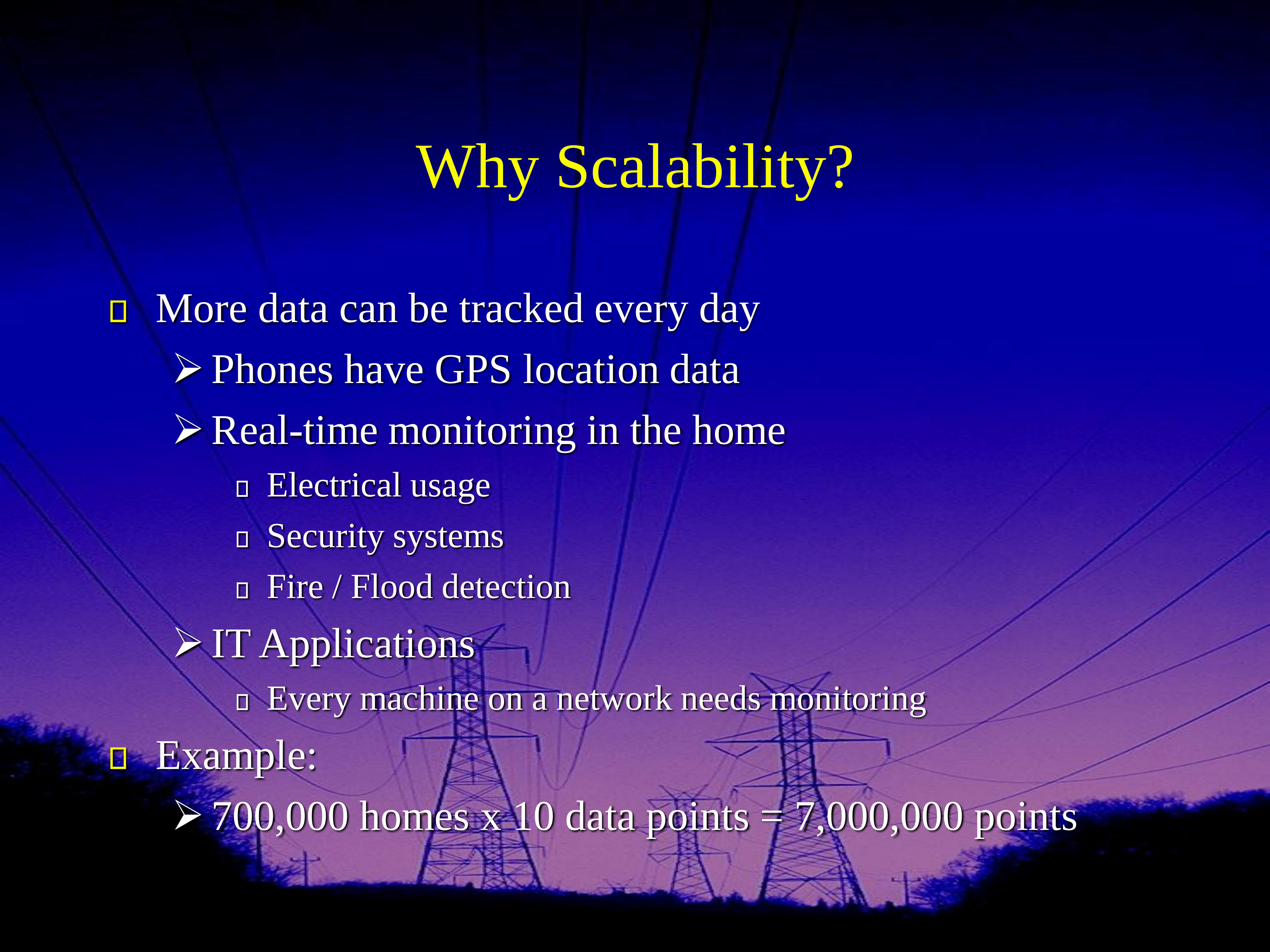
OSIsoft Overview

Developments in the PI System
Family of Products

Gregg Le Blanc

PI System Product Manager

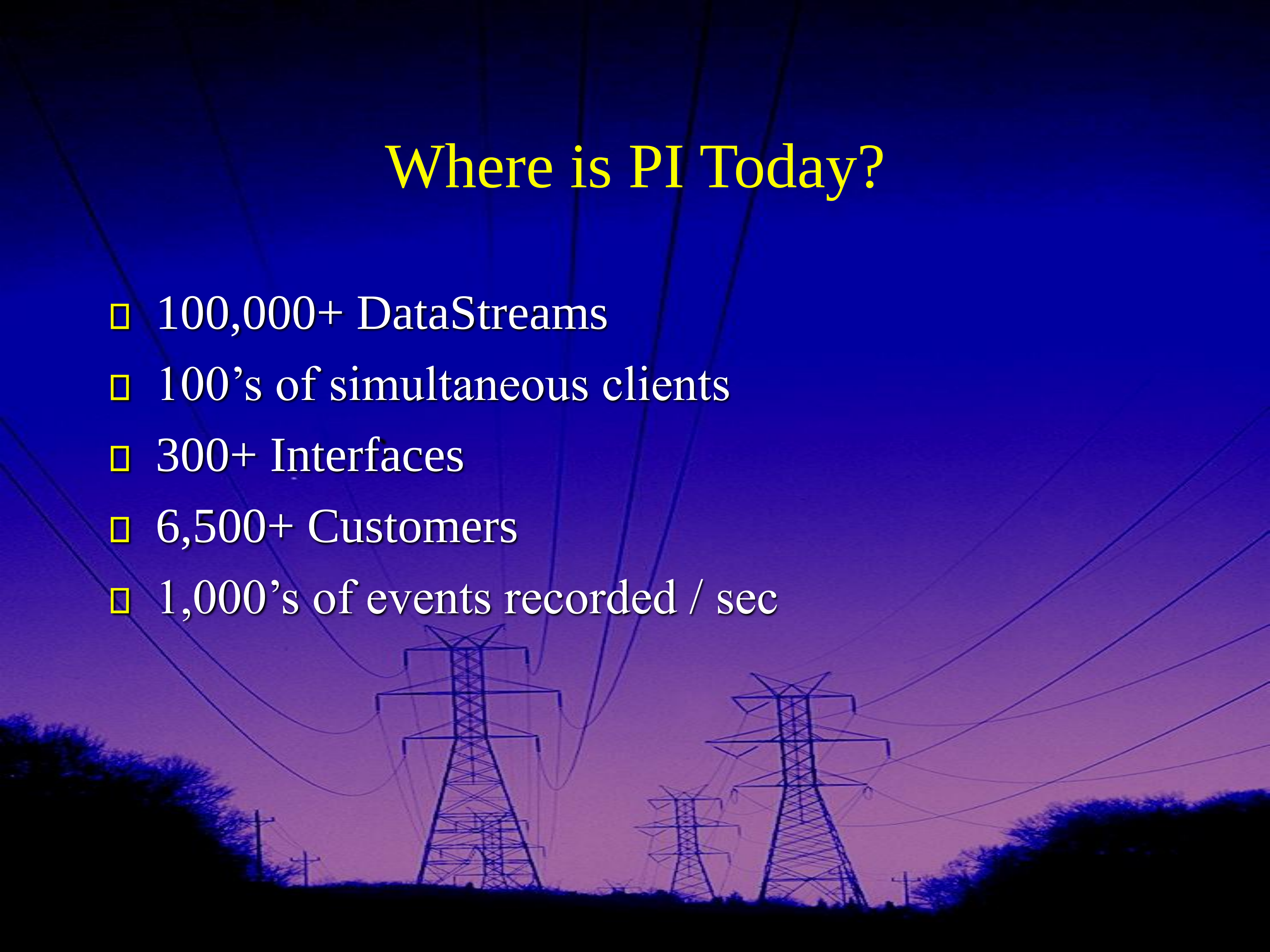


Why Scalability?

- More data can be tracked every day
  - Phones have GPS location data
  - Real-time monitoring in the home
    - Electrical usage
    - Security systems
    - Fire / Flood detection
  - IT Applications
    - Every machine on a network needs monitoring
- Example:
  - 700,000 homes x 10 data points = 7,000,000 points

Benefits

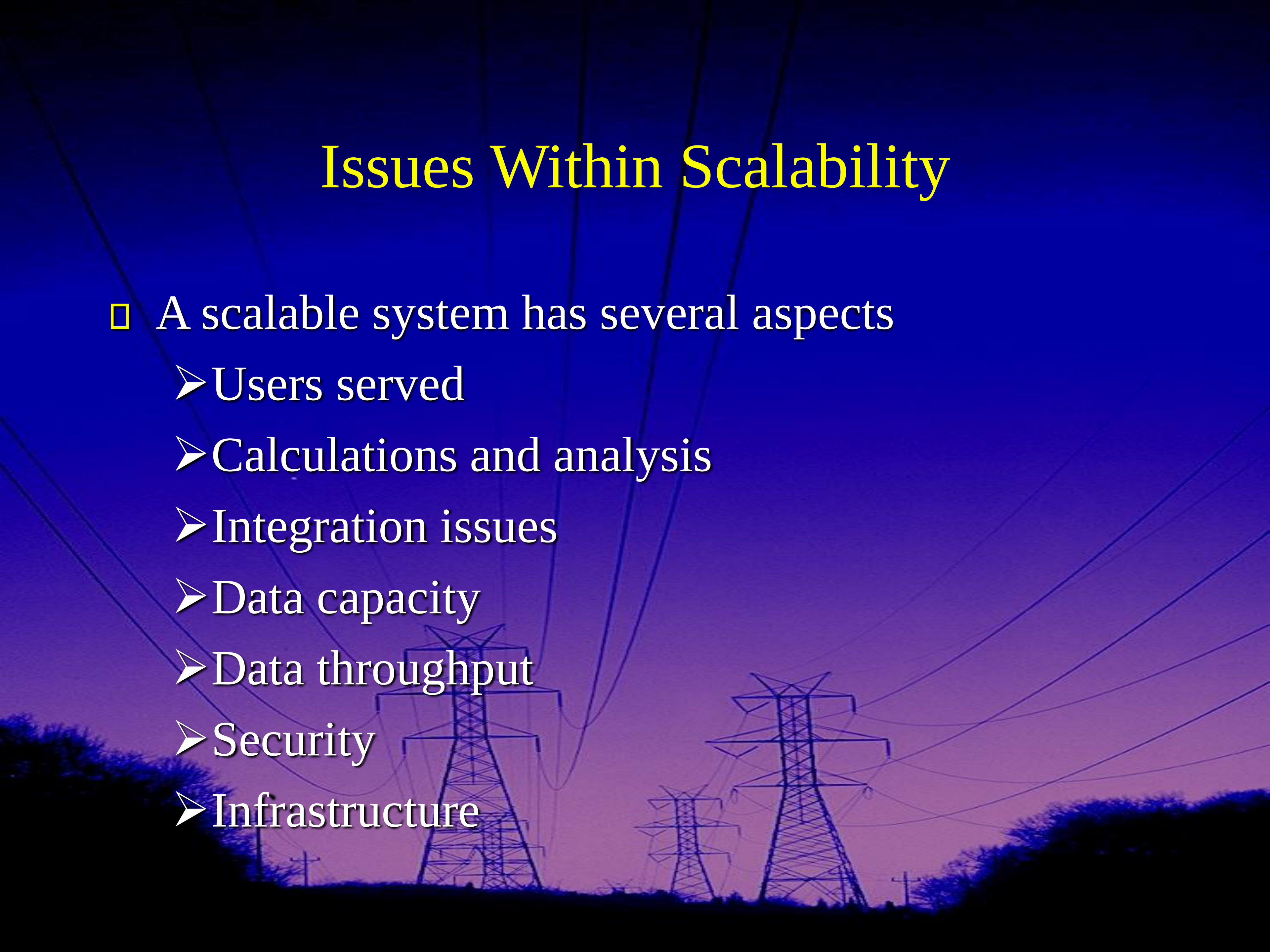
- IT Monitoring can detect
  - Intrusions
  - Impending hardware problems
  - Diagnose software problems
  - Virus traffic
- Example:
  - Because WiredCity could detect and record the amount of inbound traffic from the Nimda worm, they got a refund from MCI Worldcom



Where is PI Today?

- 100,000+ DataStreams
- 100’s of simultaneous clients
- 300+ Interfaces
- 6,500+ Customers
- 1,000’s of events recorded / sec

Issues Within Scalability

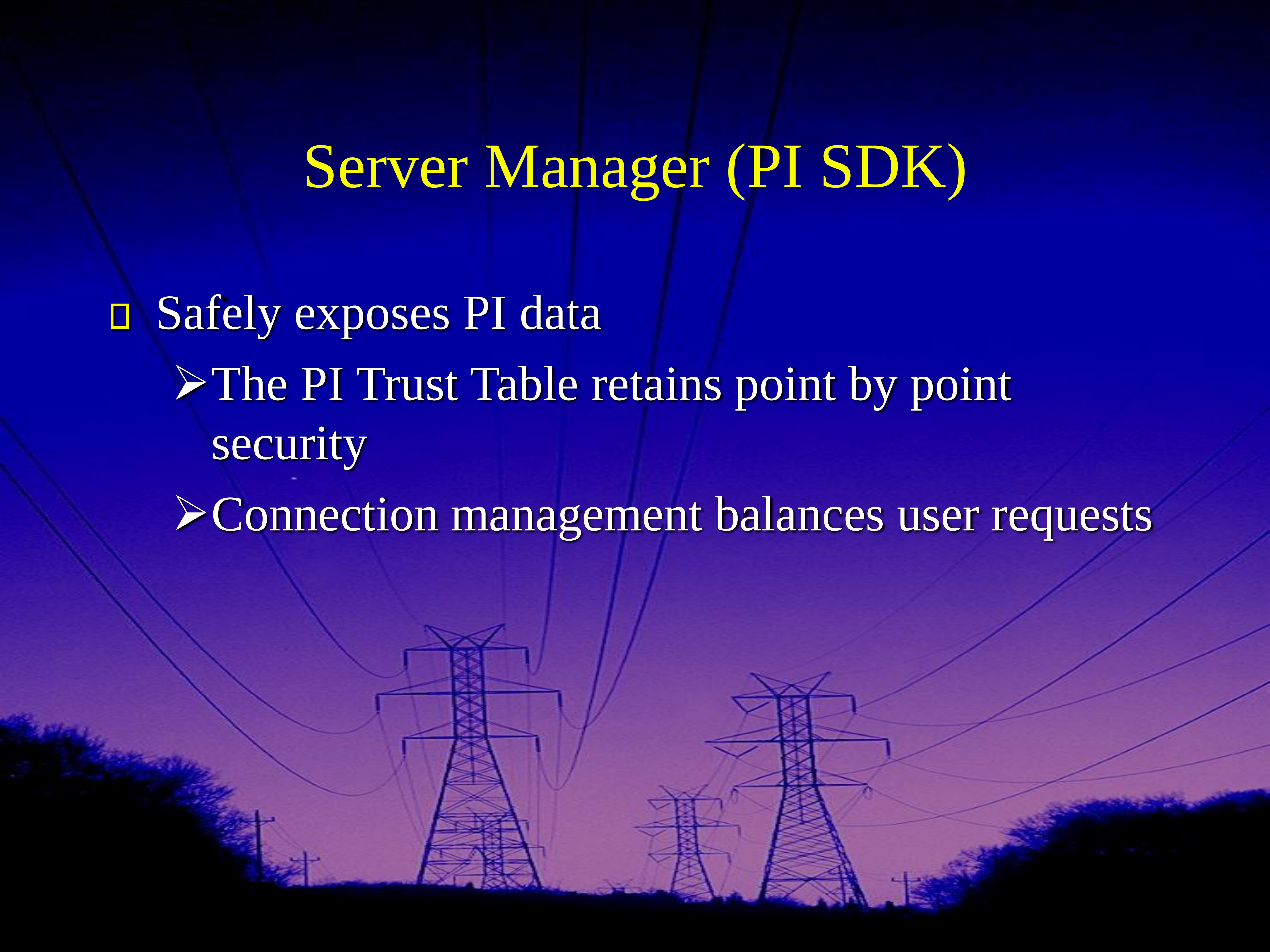


- A scalable system has several aspects
  - ➢Users served
  - ➢Calculations and analysis
  - ➢Integration issues
  - ➢Data capacity
  - ➢Data throughput
  - ➢Security
  - ➢Infrastructure

User Load & Integration

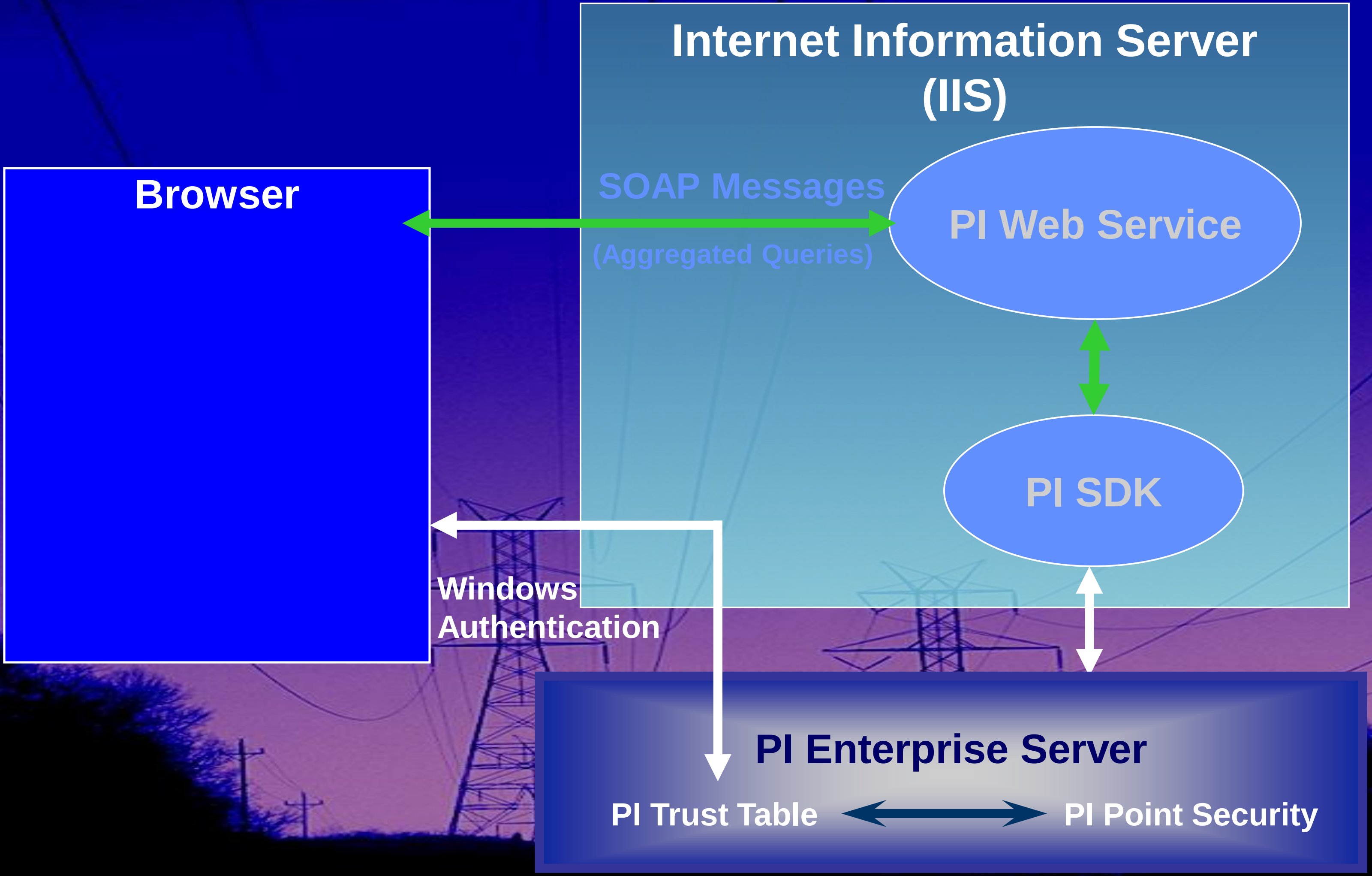
- Users need fast access to the data
  - Client / Server users
  - Stateless Users
- Client / Server users connect directly to PI
- Stateless Users
  - Broader audience
  - PI ICE
  - Web services and integration

Server Manager (PI SDK)

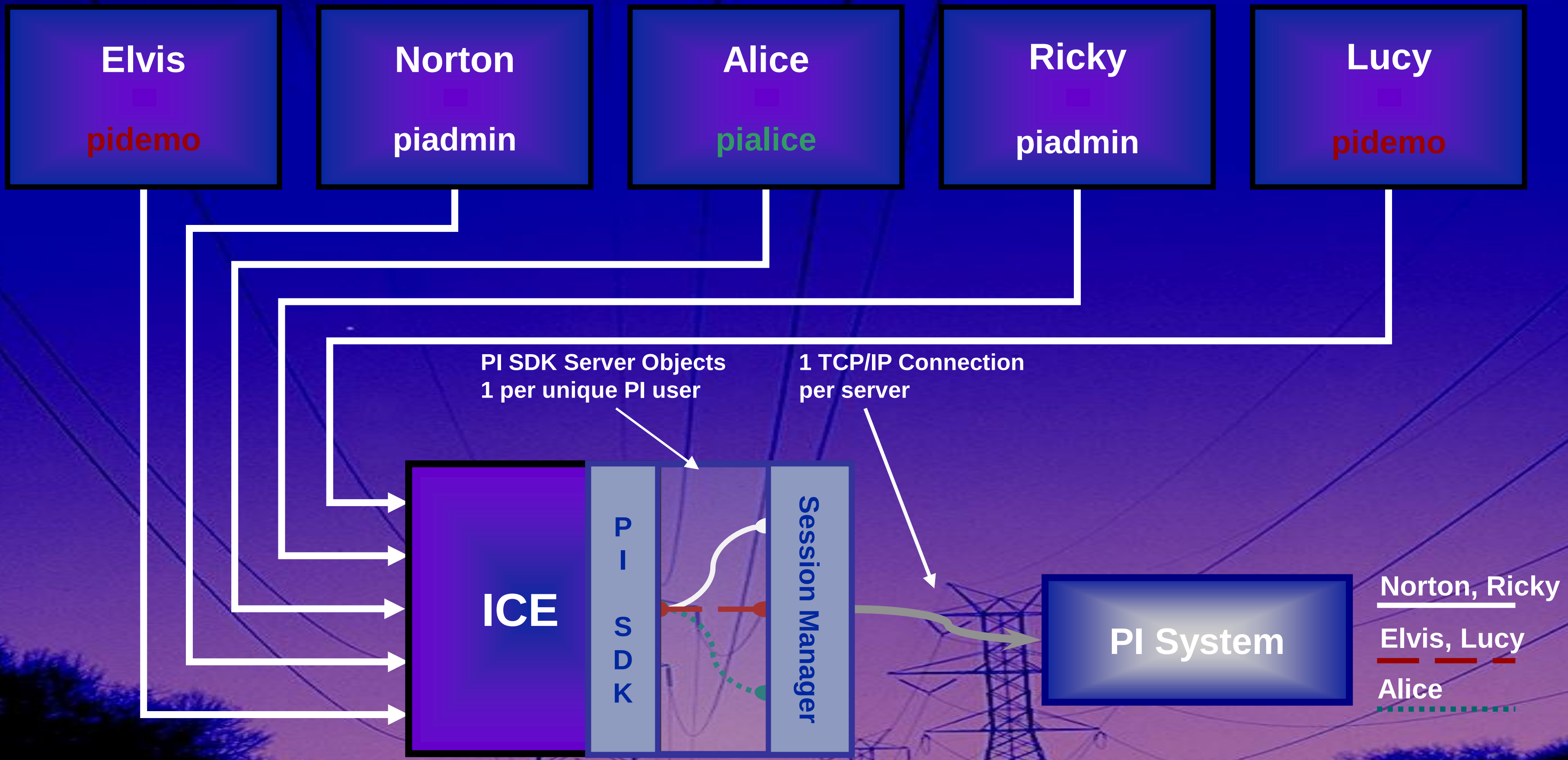


- Safely exposes PI data
  - The PI Trust Table retains point by point security
  - Connection management balances user requests

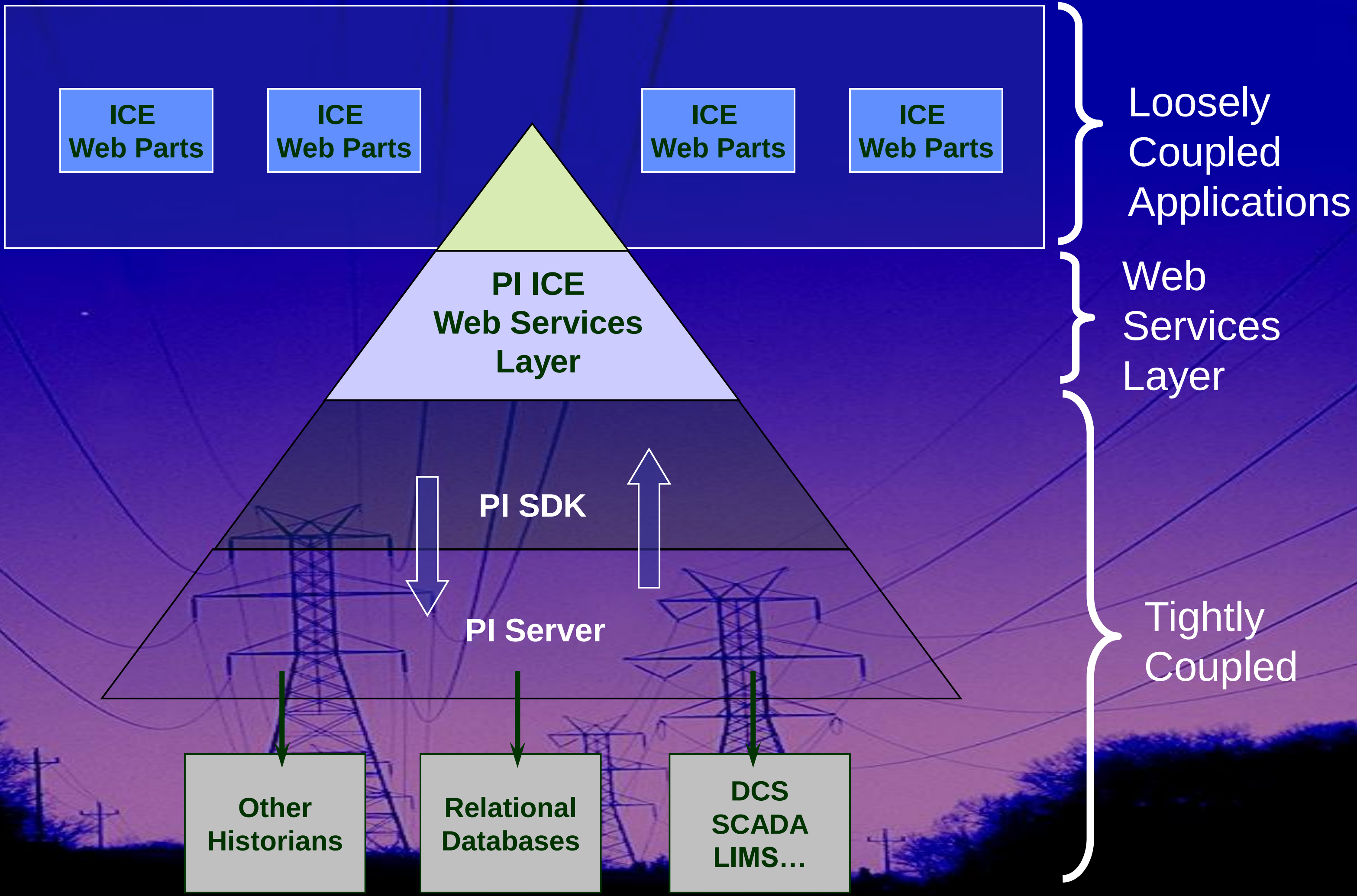
Security in ICE



Connections in ICE



Underlying Web Service Structure



Loosely Coupled Computing

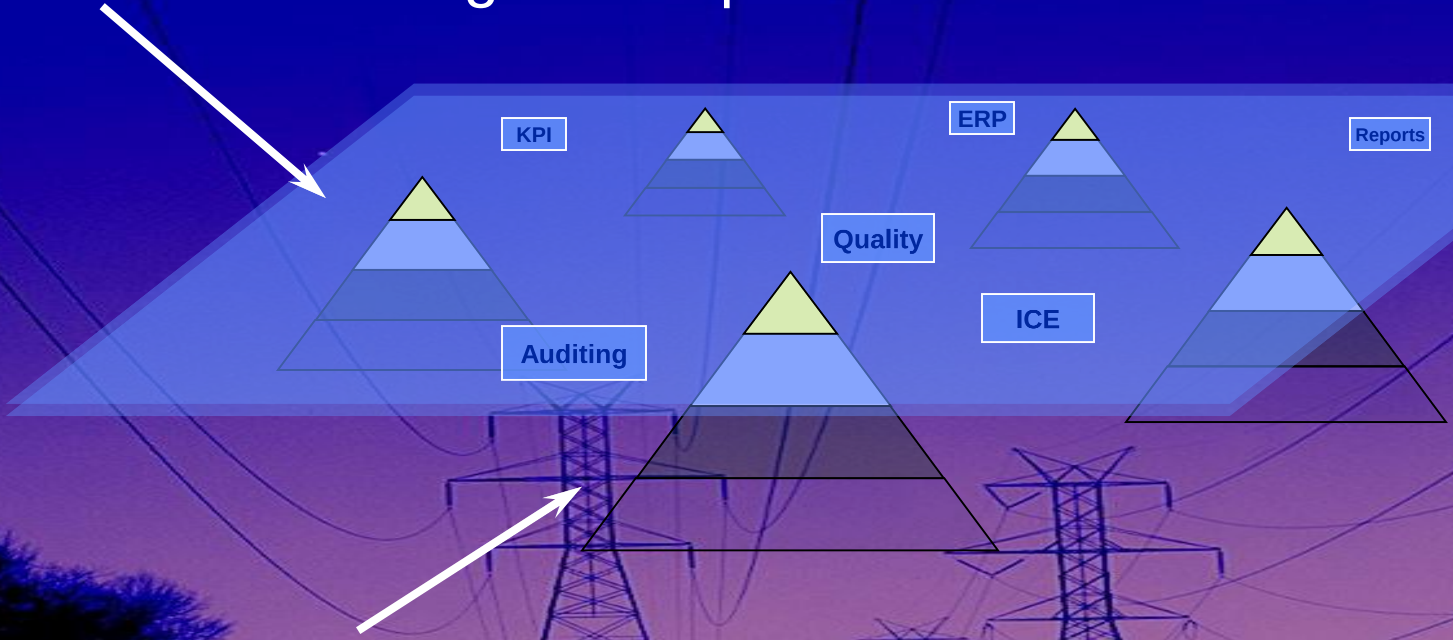
- The idea of Web services
- Allows for computing in an environment that is:
  - Asynchronous
  - Stateless
  - Platform independent
  - Geographically independent
- Will replace some “tightly coupled” computing

Tightly Coupled Computing

- Infrastructure-level ties between systems
  - Uses API calls
  - Usually proprietary
  - Not as flexible
- Examples:
  - PI Interfaces to other information systems
  - OLEDB / ODBC connectors
  - Point-to-Point, Middleware, and other integration

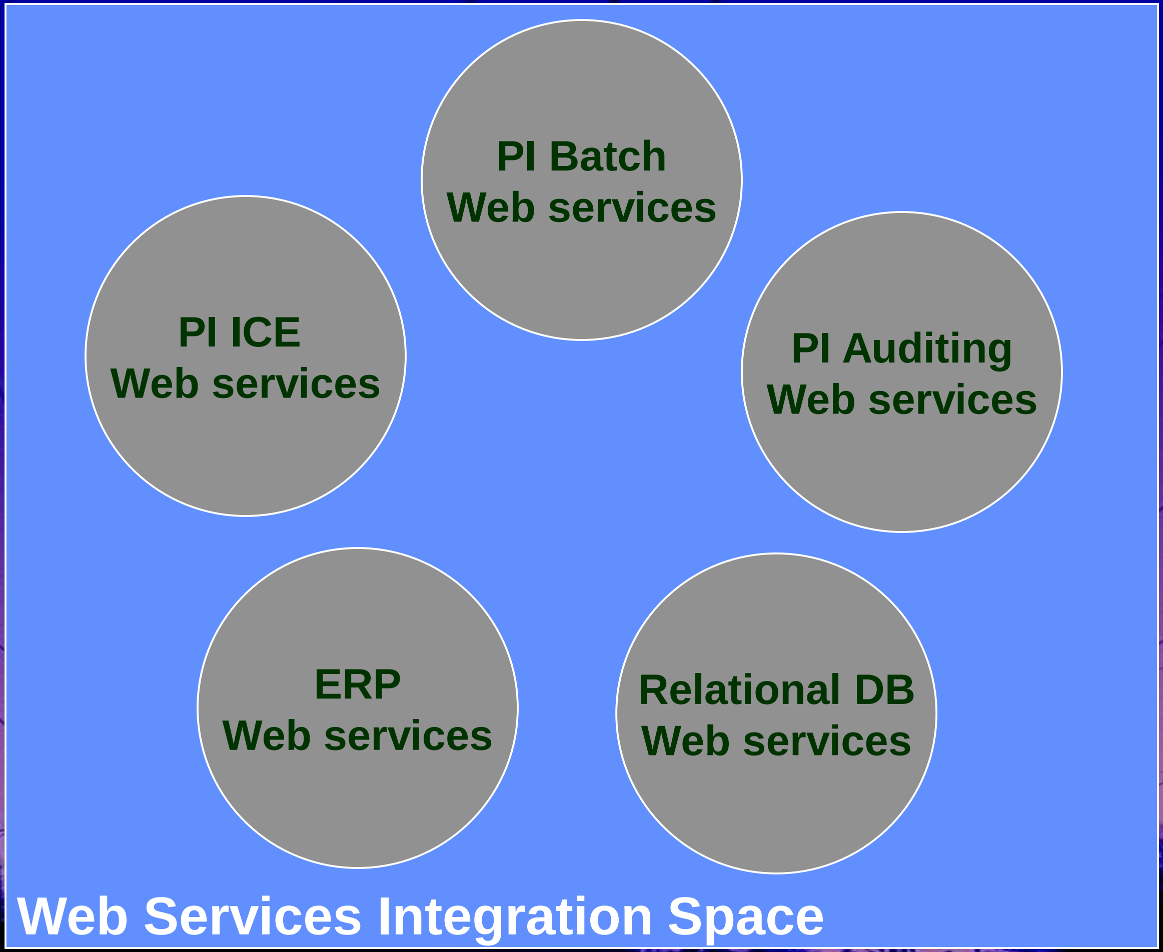
Web Service Integration

Loosely coupled applications built on Web Service Integration Space



Traditional Tightly Coupled Integration Space

Web Services



PI Batch
Web services

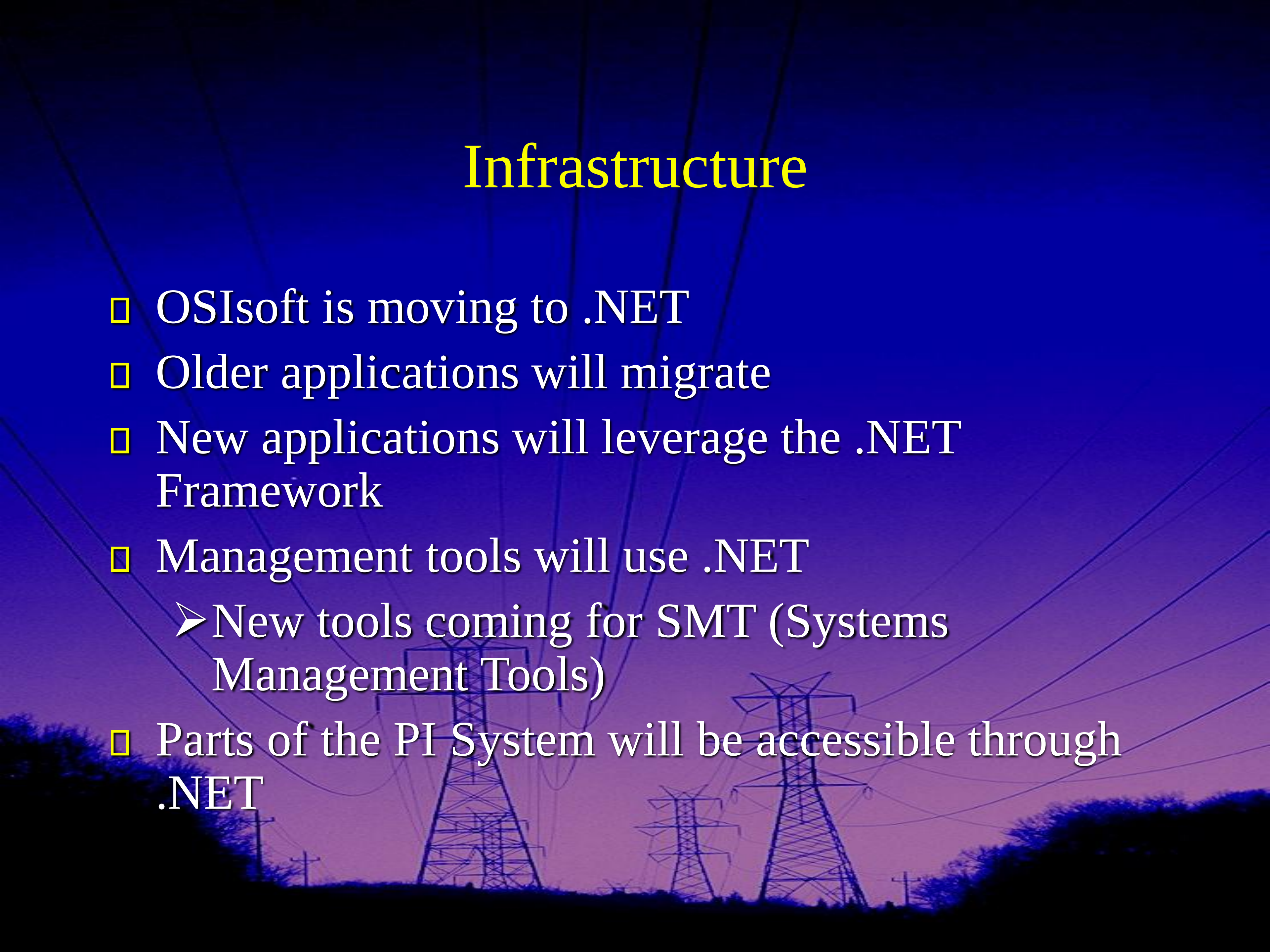
PI ICE
Web services

PI Auditing
Web services

ERP
Web services

Relational DB
Web services

Web Services Integration Space



Infrastructure

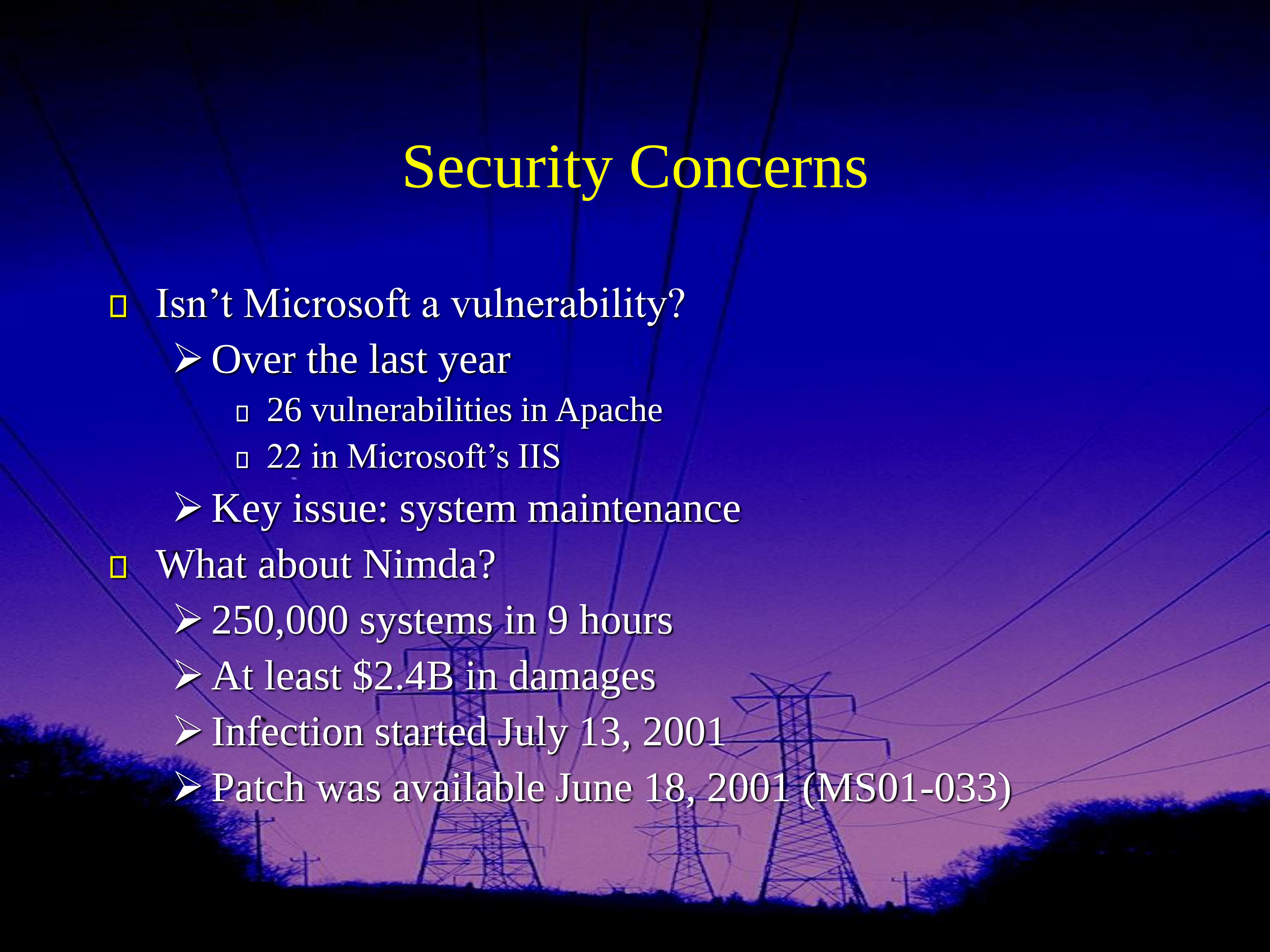
- OSIsoft is moving to .NET
- Older applications will migrate
- New applications will leverage the .NET Framework
- Management tools will use .NET
  - New tools coming for SMT (Systems Management Tools)
- Parts of the PI System will be accessible through .NET

Distributed Analysis

- Server Based
  - Performance Equations
- Desktop Based
  - Spreadsheets based on DataLink
- Distributed Calculations
  - PI ACE
- Server load reduced
- Calculations can be reused

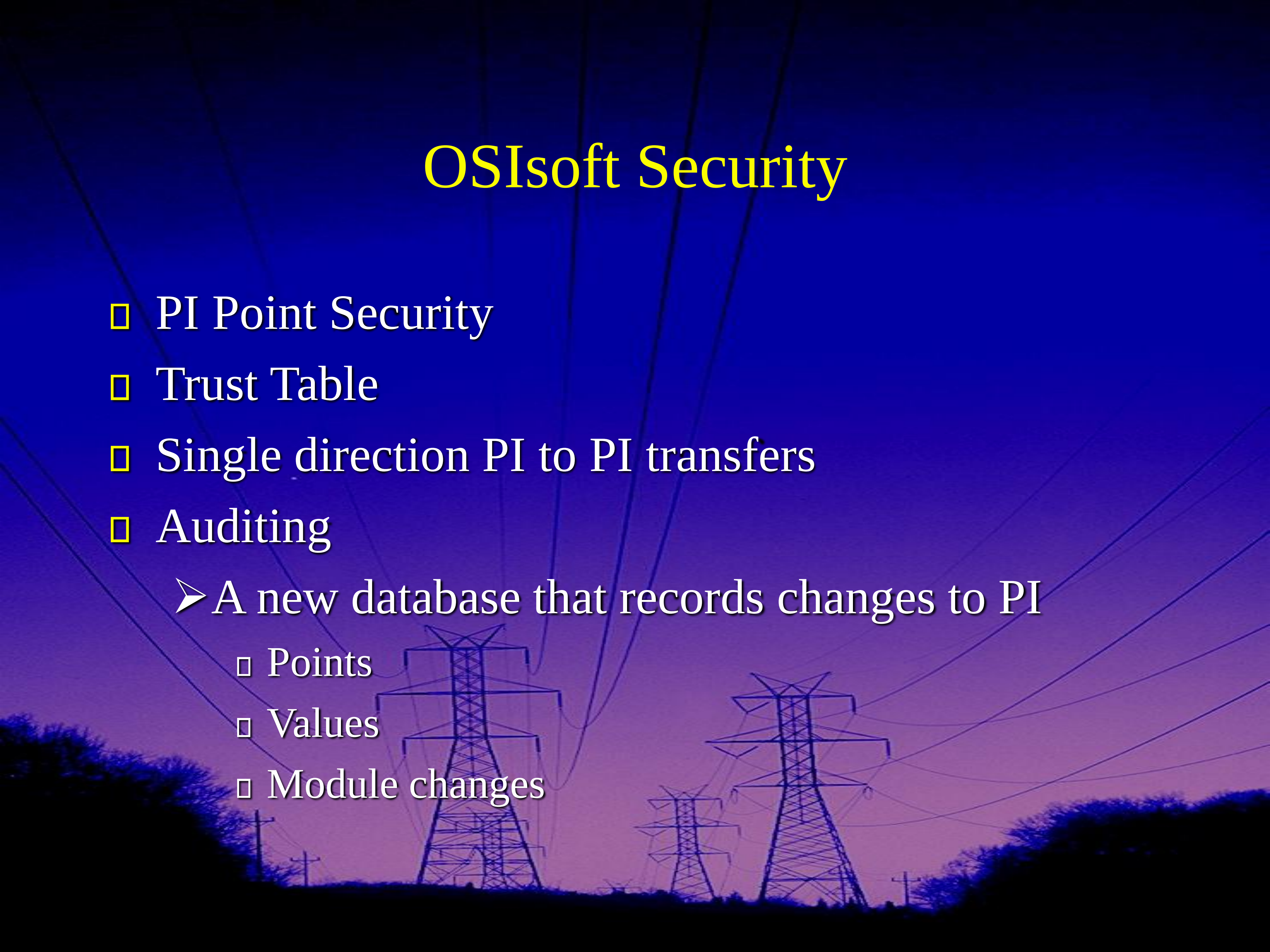
Security Concerns

- Isn’t Microsoft a vulnerability?
  - Over the last year
    - 26 vulnerabilities in Apache
    - 22 in Microsoft’s IIS
  - Key issue: system maintenance
- What about Nimda?
  - 250,000 systems in 9 hours
  - At least $2.4B in damages
  - Infection started July 13, 2001
  - Patch was available June 18, 2001 (MS01-033)



Security Response

- More accountability: Microsoft
- NIPC (National Infrastructure Protection Center) works with NERC and Microsoft to develop security procedures and standards
- Key precautions still remain
  - Guard against social engineering
  - System maintenance
  - Unknown vulnerabilities
    - WiFi, wardriving
    - Unprotected modem lines used for support



OSIsoft Security

- PI Point Security
- Trust Table
- Single direction PI to PI transfers
- Auditing
  - A new database that records changes to PI
    - Points
    - Values
    - Module changes

What’s Been Covered

- Handling user load
- Calculations
- Integration demand
- Security
- Infrastructure
- Million point PI Systems

History of the Historian

- PI 1
  - 1983 HP-1000
  - 1985 Vax-VMS
  - 1988 PINet
    - Client/Server Architecture
- PI 2
  - 1992 Vax-VMS
  - 1994 Alpha-OpenVMS
- PI 3
  - 1993 Design Work

History of the Historian

- PI 3
  - Big 4 Unix
    - HP-UX
    - IBM-AIX
    - Dec-OSF/1
    - Sun-Solaris
  - Windows NT 3.51
  - Development Language C++

History of the Historian

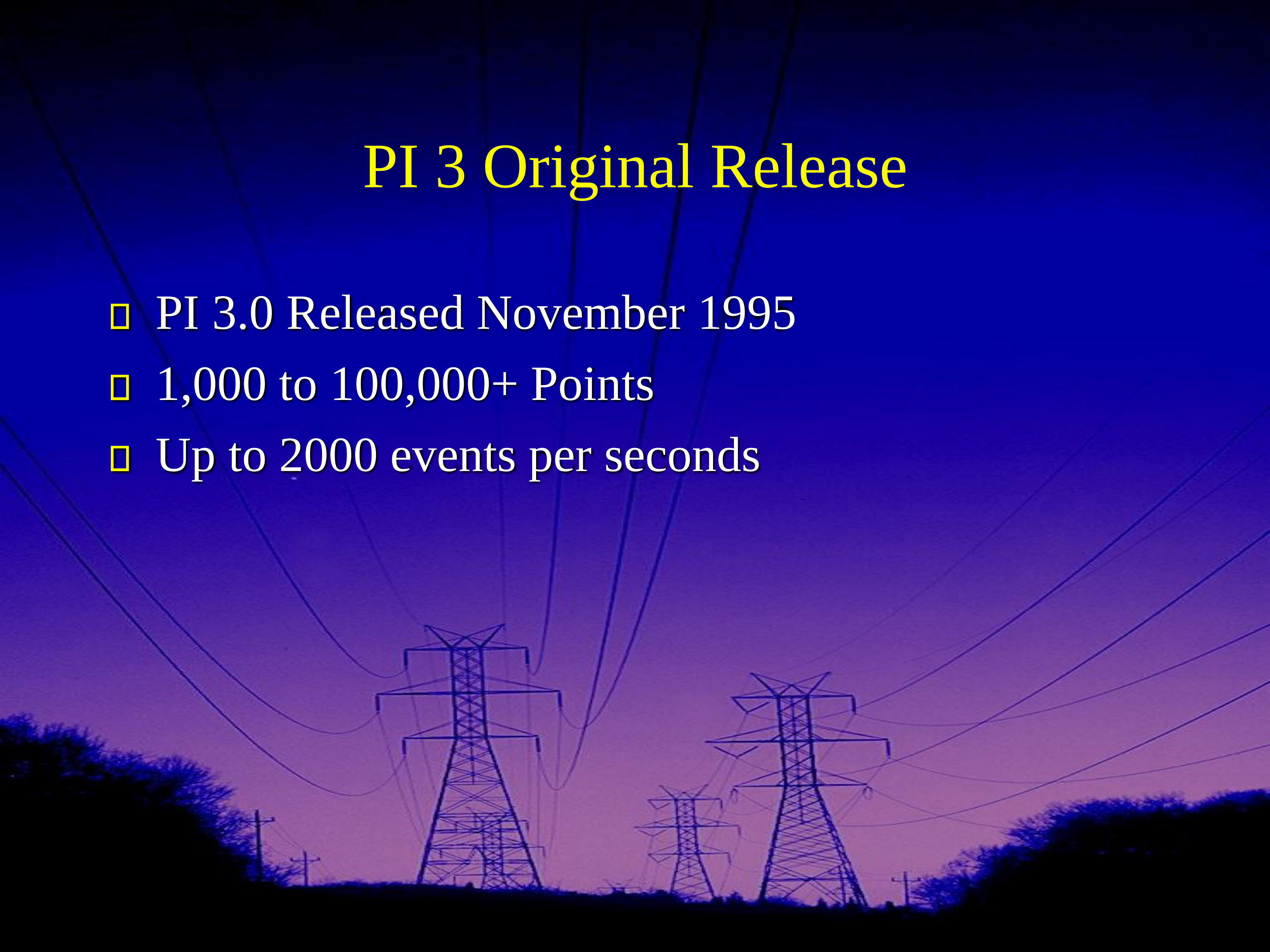
- PI 3
  - Key concepts of PI 2 were used
    - Snapshot
    - Compression
    - Archive cache
    - Archive navigation
    - PINet

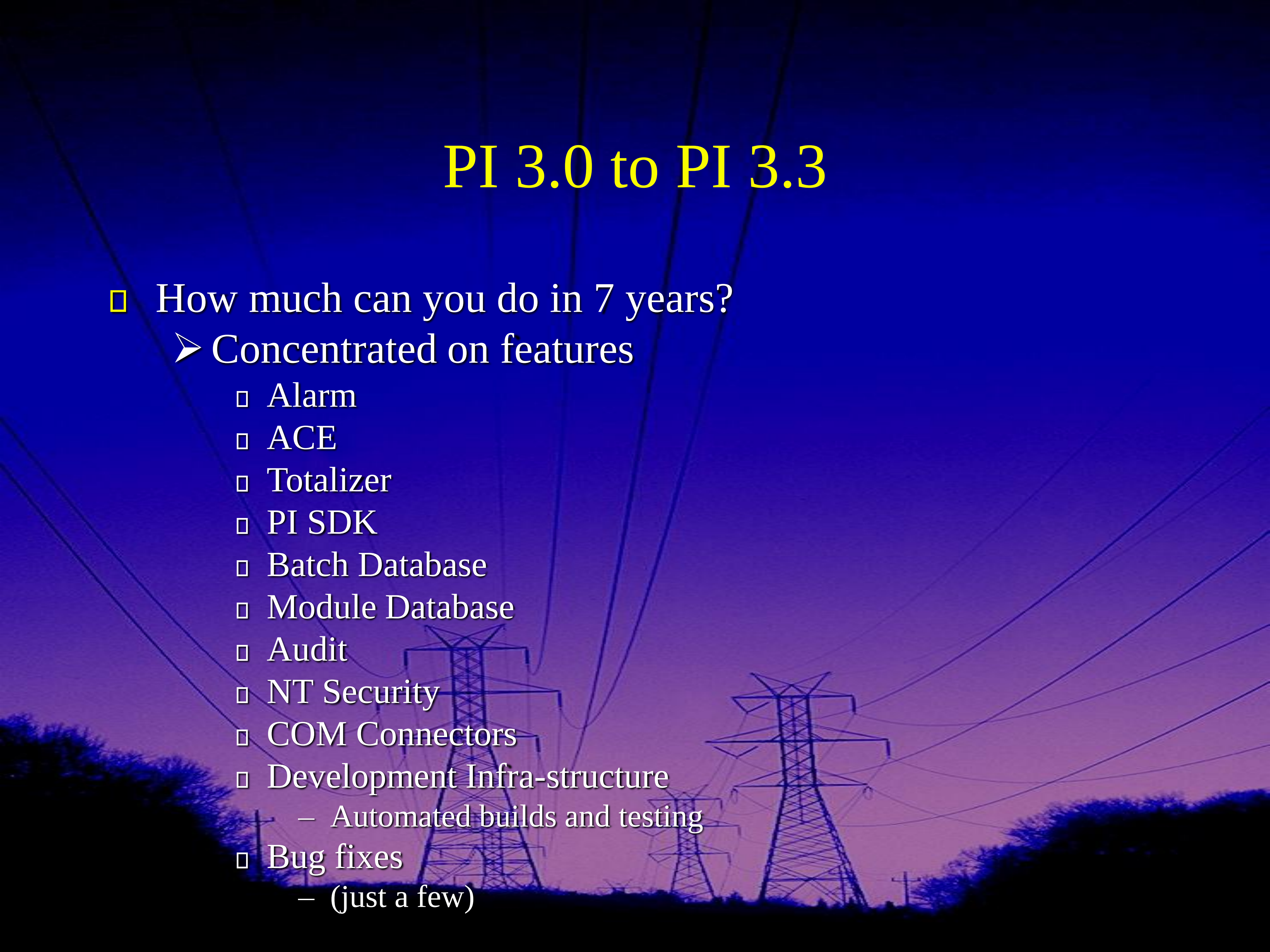
History of the Historian

- PI 3
  - Key new concepts
    - Multi-process
    - RPC Based Inter-process communication
      - PI SDK
    - Historization of many different data types
      - Doubles
      - Strings
      - BLOBs
    - Common code base for all platforms
      - 64 bit support

PI 3 Original Release

- PI 3.0 Released November 1995
- 1,000 to 100,000+ Points
- Up to 2000 events per seconds

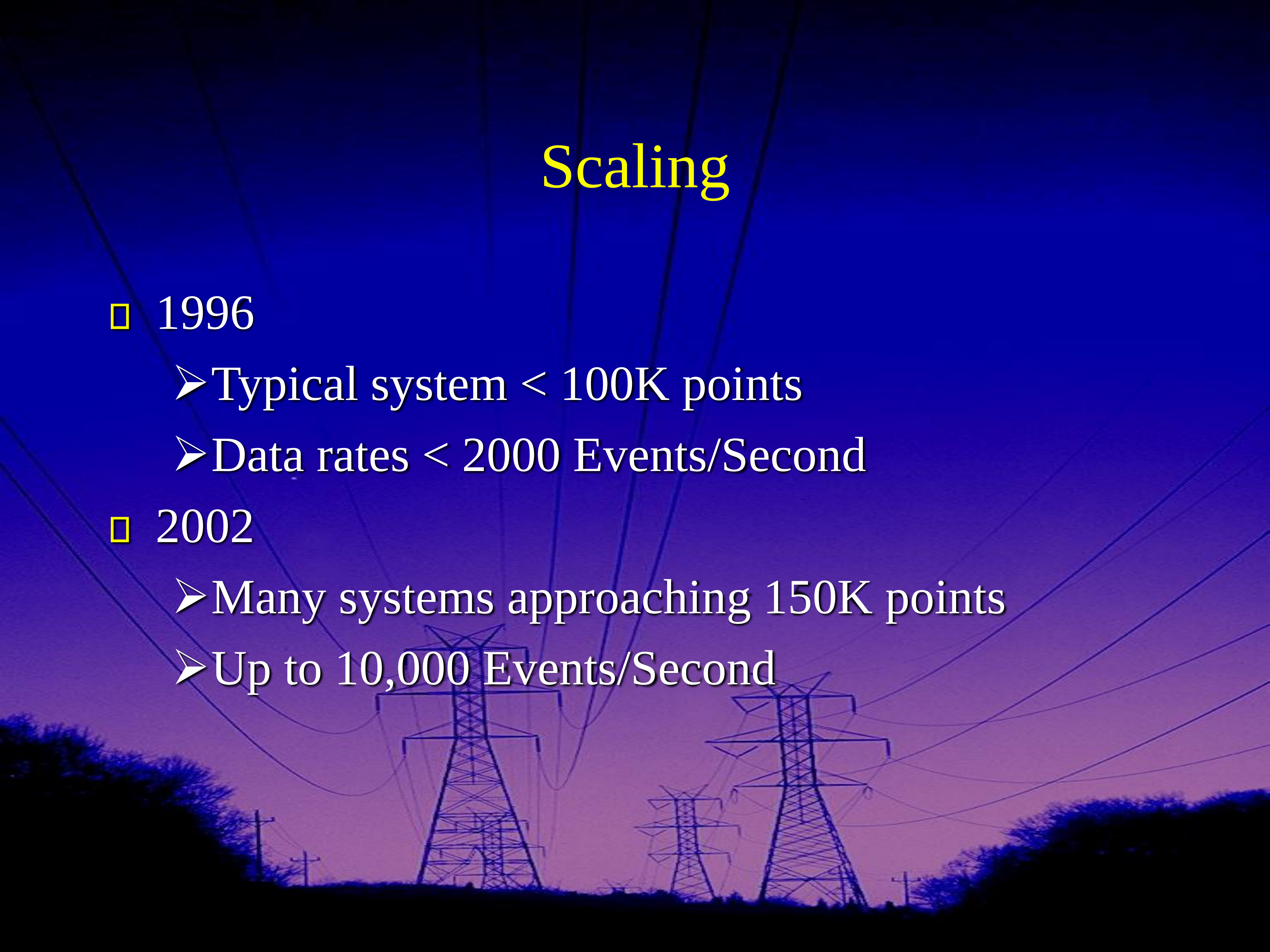




PI 3.0 to PI 3.3

- How much can you do in 7 years?
  - Concentrated on features
    - Alarm
    - ACE
    - Totalizer
    - PI SDK
    - Batch Database
    - Module Database
    - Audit
    - NT Security
    - COM Connectors
    - Development Infra-structure
      - Automated builds and testing
    - Bug fixes
      - (just a few)

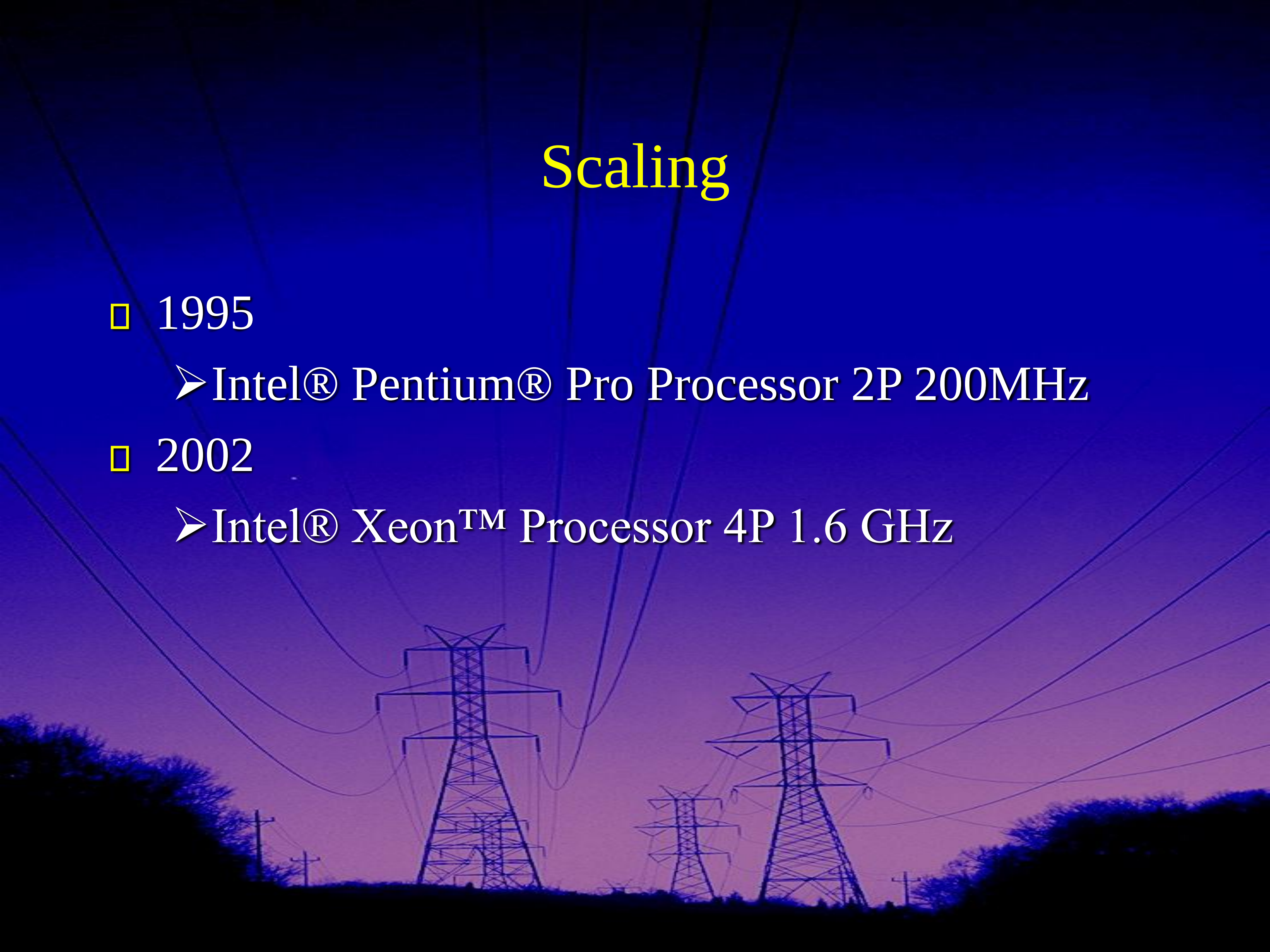
Scaling

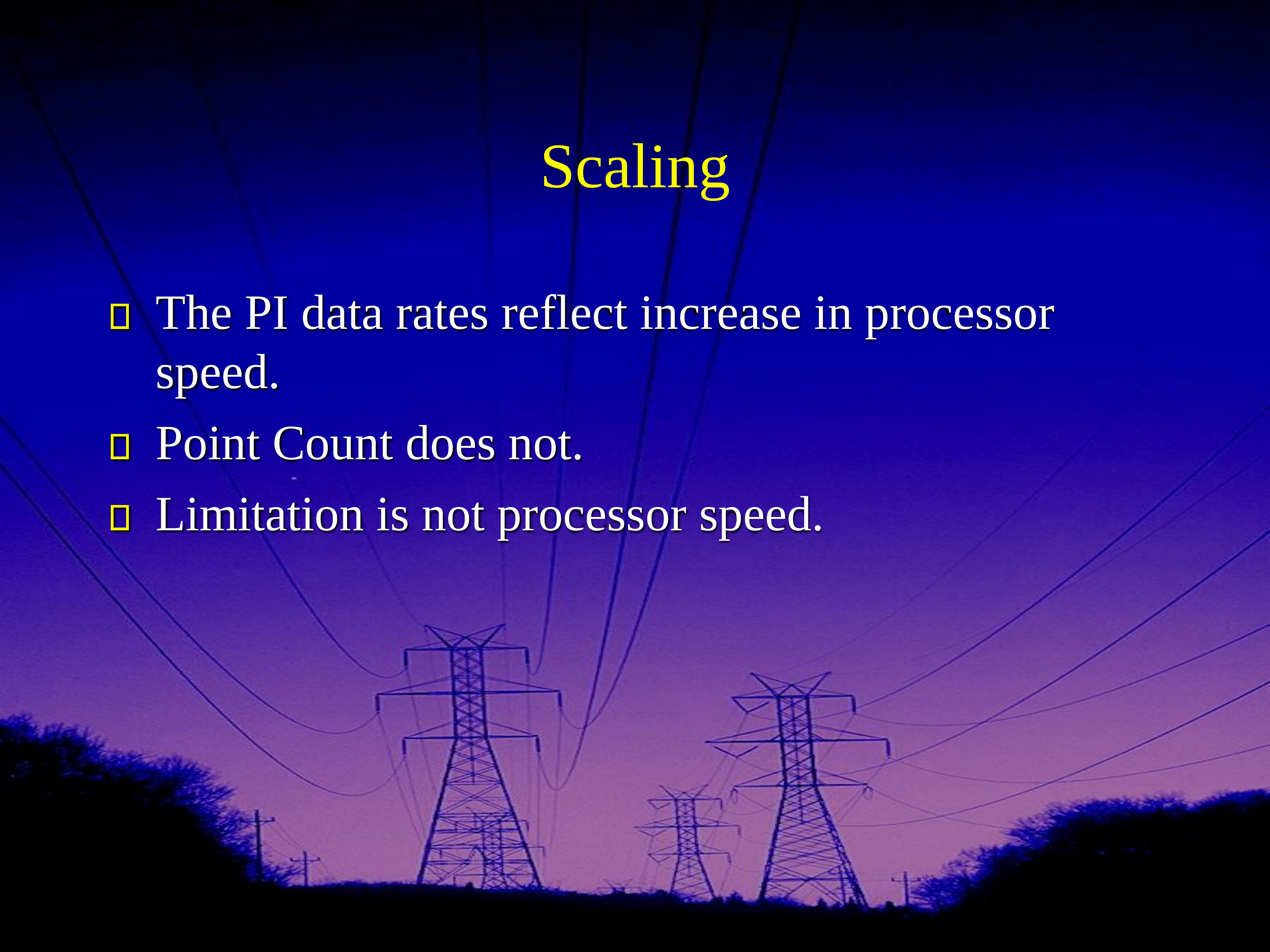


- 1996
    - Typical system < 100K points
    - Data rates < 2000 Events/Second
- 2002
    - Many systems approaching 150K points
    - Up to 10,000 Events/Second

Scaling

- 1995
  - Intel® Pentium® Pro Processor 2P 200MHz
- 2002
  - Intel® Xeon™ Processor 4P 1.6 GHz





Scaling

- □ The PI data rates reflect increase in processor speed.
- □ Point Count does not.
- □ Limitation is not processor speed.

Limitations in PI 3.3

- Memory
  - 2 GB of virtual memory per process
    - Archive cache
- Inter-process communication
  - Compressed events from snapshot to archive
- Serialization of RPCs
  - Archive sub-system handles one call at a time

Memory

- PI Databases are memory resident
  - Point database
  - Module database
  - Snapshot
- PI Archive Cache

PI 3.3 Cache Memory Issues

- Cache record can be quite large.
- Adding a single event requires entire cache record
- Systems receiving data for most points will have a large memory footprint
  - Lots of cache activity
    - Pushing records out of memory
    - Reading records into memory

PI 3.4 Archive Cache

- Typical PI System
  - Majority of points receive data regularly
    - Add events
    - New events received in order, and near current time

The PI 3.4 Cache

- Addresses Memory Issues
  - Smaller memory footprint
  - More efficient handling of new events
  - Same efficiency reading data
- Design Tradeoffs
  - More complicated

3.3 Snapshot – Archive
Inter-process Communication

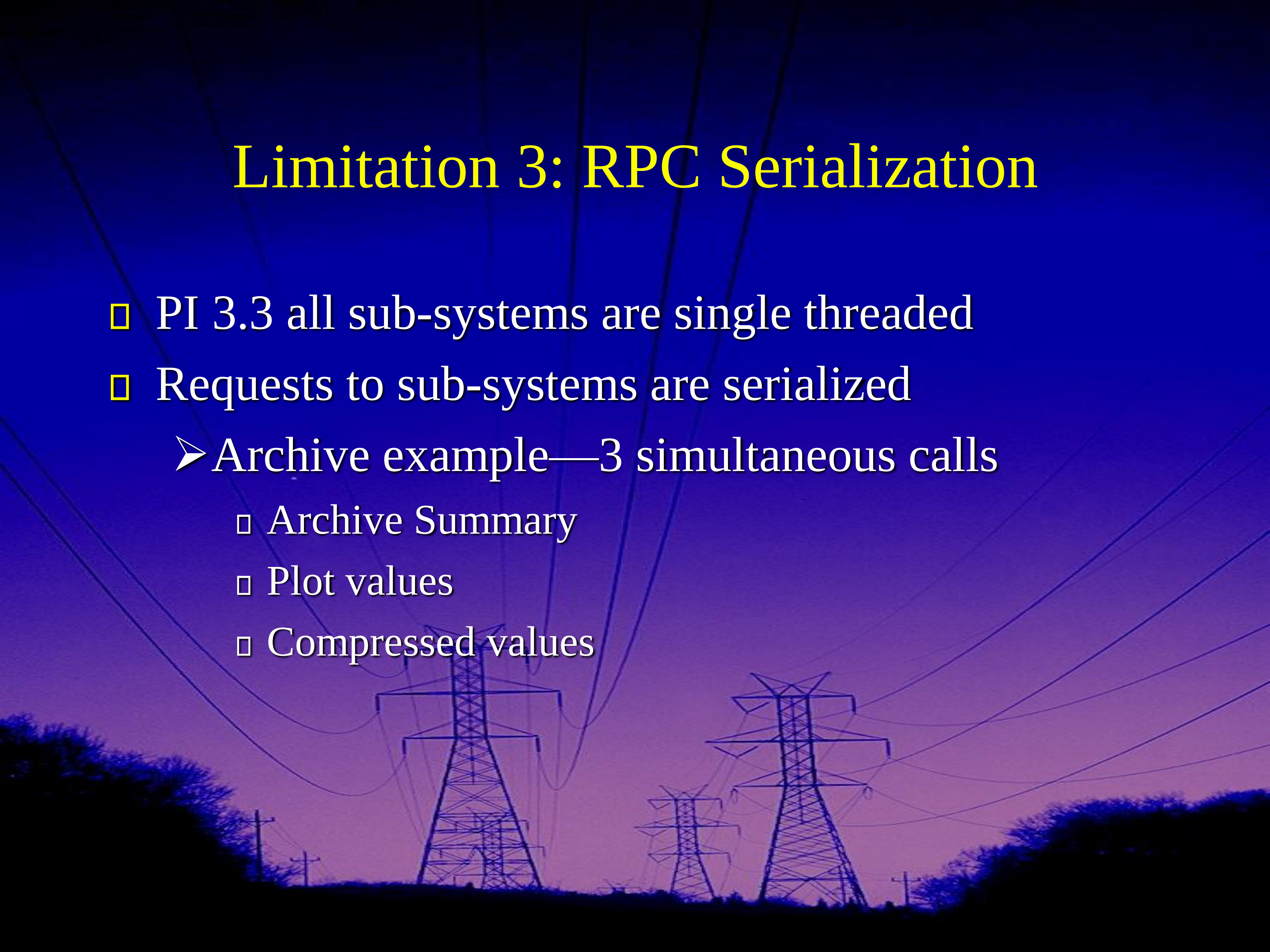
- Current Mechanism
  - Event received is processed by the compression algorithm.
  - If passes compression event is added to in-memory event queue
  - Events in queue are packaged and sent via an RPC to the archive process

3.3 Snapshot Buffering

- Archive unable to process compressed events
  - Backups
  - Extremely high data rates
  - System problems
- Snapshot must write to “event queue” file
- Archive must read from the Snapshot event queue
- Result
  - 2 physical disk I/O’s happen on same file

3.4 Snapshot – Archive Inter-process Communication

- Changed physical file to memory mapped file
  - File system feature
- Mapped by Snapshot and Archive
- Snapshot writes compressed events to “file”
- Archive reads compressed events from “file”
  - Synchronization techniques required to prevent corruption
- Buffering case handled by file system



Limitation 3: RPC Serialization

- PI 3.3 all sub-systems are single threaded
- Requests to sub-systems are serialized
  - Archive example—3 simultaneous calls
    - Archive Summary
    - Plot values
    - Compressed values

RPC Serialization Solution

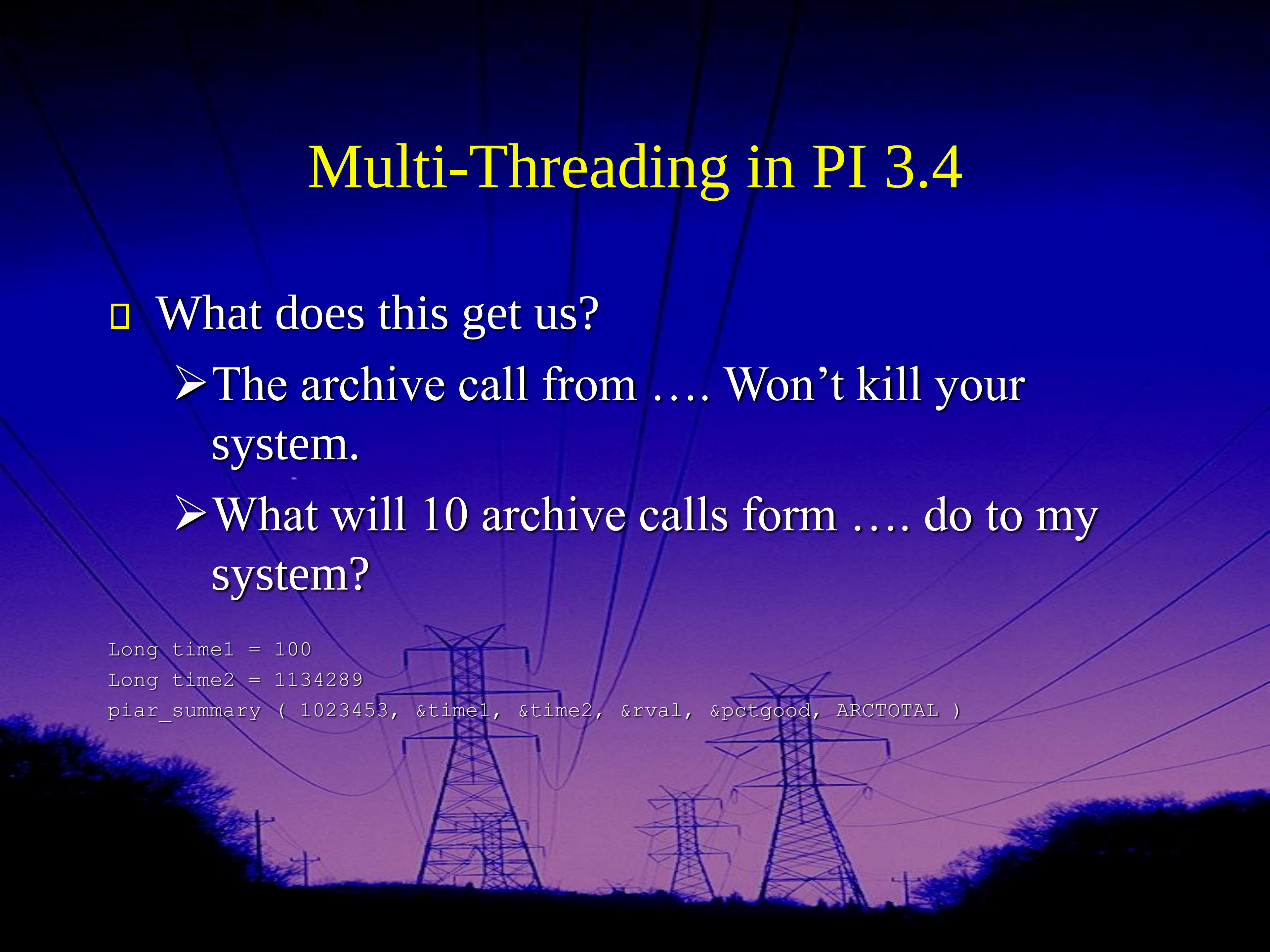
- Multiple Threads
  - Threads are the obvious approach
  - Unix and NT have very powerful and easy to use threading models
- Conceptually Simple
- But, easy to write bugs

Multi-Threading in PI 3.4

- PI 3 had some threading since the original release
  - PI Net Manager
  - Sub-systems use a thread for reading messages
- PI 3.4 Sub-system Level thread model
  - Main “house keeping” thread
  - Read thread
  - Message pump
  - Pool of worker threads

Multi-Threading in PI 3.4

- PI 3.4 Sub-system Level thread model
  - Every sub-system is multi-threaded
  - Sub-system must implement locks for optimal performance
  - Worker thread pool is configurable
    - Runtime
      - Kill threads
      - Suspend threads
      - Change priority
      - Add worker threads
      - Delete worker threads



Multi-Threading in PI 3.4

- What does this get us?
  - The archive call from …. Won’t kill your system.
  - What will 10 archive calls form …. do to my system?

```
Long time1 = 100
Long time2 = 1134289
piar_summary ( 1023453, &time1, &time2, &rval, &pctgood, ARCTOTAL )
```

PI on 64 Bit Windows

- One advantage: larger addressable memory
  - 32 bit: 2 GB
  - 64 bit: 16 – 128 GB
    - XP-64: 16 GB
    - .Net Enterprise Server: 64 GB
    - .Net Data Center: 128 GB

PI on 64 Bit Windows

- Disadvantages:
  - Slower
  - Very sensitive to properly optimized code
- Servers only
  - 64 bit Windows is not intended to replace 32 bit desktops.
  - Only makes sense to port server applications
    - 64 bit development will be done on 32 bit machines

PI on 64 Bit Windows

- PI 3 is designed for 64 bit
  - Supports Dec OSF/1
    - Original release of PI supported Dec’s 64 bit Unix and still does—HP Tru64 Unix.
- PI 3.4 will be 64 bit ready
  - Minor differences between Windows and Unix 64 bit
  - 3.4 will address these minor issues
  - 3.4 will compile cleanly

Summary

- Point Count limited by archive cache memory
- 3.4 archive cache is significantly smaller and more efficient
- Memory mapped files for configuration data
  - Point Database
- Tested million point system on typical 32 bit server class machine
  - 2P 1 GHz Pentium 4 / 4 GB Memory
  - Several million points possible
- 64 bit, perhaps 50 Million points?

Summary

- Snapshot – Archive inter-process communication
  - Significantly faster the RPC mechanism
  - Memory mapped file shared by both processes
  - Backed by disk to insure no data loss
- RPC Serialization
  - Eliminated by threading
  - Process several calls simultaneously

Questions?