

# StrongKey

The industry's first open-source SKMS

Arshad Noor  
CTO, StrongAuth, Inc.

IEEE Key Management Summit 2008

# Who is StrongAuth, Inc.

- 7+ year-old, private company based in Sunnyvale, California
- Focus on:
  - Enterprise Key Management
  - PKI-based Identity Management
- Products/Services:
  - PKI Appliance, StrongKey, CSRTool
  - Managed PKI Services
- Founding member of OASIS EKMI TC

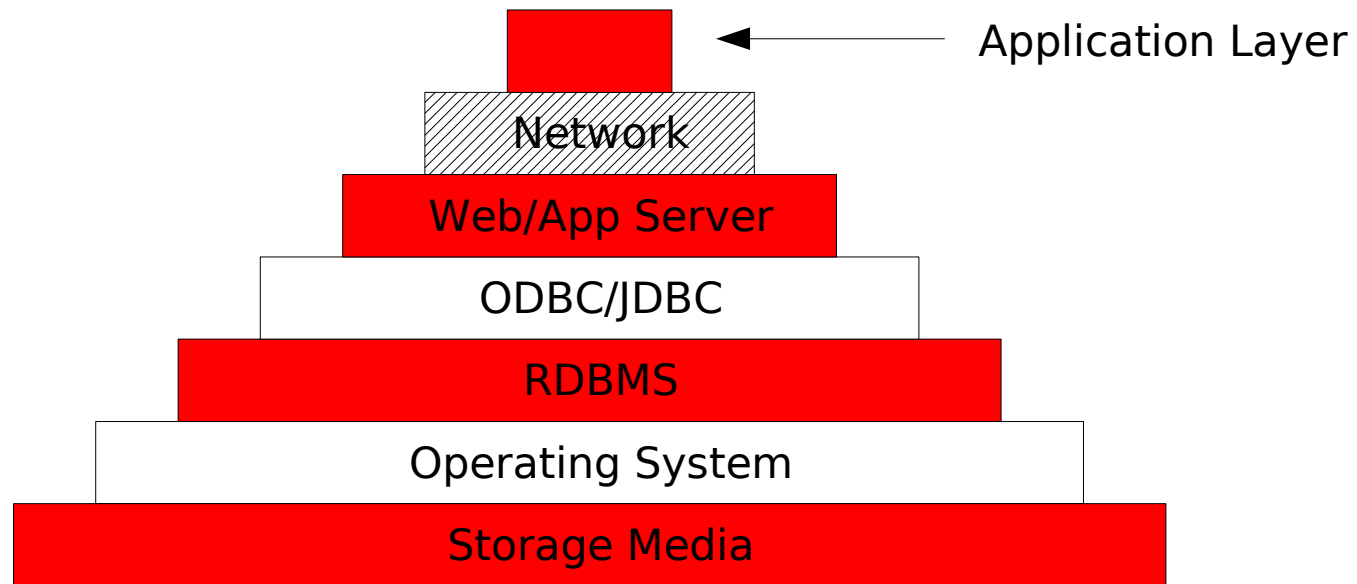
- public Symkey getSymkey (.....)
- public byte[] encryptToBytes (.....)
- public String encryptToXML (.....)
- public byte[] decryptBytes (.....)
- public byte[] decryptXML (.....)
- public String getSHA1 (.....)
- public String getSHA256 (.....)
- public String getSHA384 (.....)
- public String getSHA512 (.....)

```
<?xml version="1.0" encoding="UTF-8"?>
<xenc:EncryptedData xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
  <xenc:EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#aes256-cbc"/>
  <ds:KeyInfo>
    <ds:KeyName>10514-1-1453</ds:KeyName>
    <ds:RetrievalMethod>http://localhost:8080/symkeyServlet/getsymkey</ds:RetrievalMethod>
    <ds:RetrievalMethod>http://skms.hospital.com/symkeyServlet/getsymkey</ds:RetrievalMethod>
  </ds:KeyInfo>
  <xenc:CipherData>
    <xenc:CipherValue>
      SID9GQ1rmuRBSlHcwK04TuGn4G8o2T99OaZtdT8BIH4rTT2YFT0a5D8FF9J0oFSJOXFLydpI8a4A
      DX7B/ZQCqa1VKhOnh0EfK0fjNfeuuw97yzTZyZ3y7uFQv1GJkqW7mBlzY9TKj4usTIOciEl1J5j1
      JwCTECcxDfb8/iKMW9GQXZzTOVjkWi+SSJSPeN1i0oZwY0o5zgVZESIV+71nDoVB2uM6p1BrM845
      jGqpuCr8gGaoD1GcF5sl2H9I4++JsmiZQZiapHvlEm3XbdsMRNsGYrVZhxbPWEvNiTuixBVJoXkx
      pY/z6pRrphzlwUCW88Ui5O7LhaBYbMepq1WJ0jyAEIaMc6LX8m8Hq+dneai2hBYSOqm0iSDInBeU
      VmkgwV7yzUBcxazr6Vdx3Z+xPI8TZtaSzffBlDcGvaExQgdOhgzKNWypUCt+NHRIBNj50XHiUcoR
      53Njd6u9Ygjl2pu48798OGmPzo7iTwd3Smj2nCoikJl2AL1xupl0hto8oBd42ItC27MDTdnSsp+
      SyeSLmnVoUwbo0DpGiwHl9pzU5leDuKG7GYWIHG7zxJHvmiRzRmOjPZIFesXxIBT0UMdugew
    </xenc:CipherValue>
  </xenc:CipherData>
  <xenc:EncryptionProperties>
    <xenc:EncryptionProperty name="InitializationVector">
      hAOPwhT0ocD4k7Jc
    </xenc:EncryptionProperty>
  </xenc:EncryptionProperties>
</xenc:EncryptedData>
```

cid	fname	lname	ssn_ciphertext	gkid
1	John	Galt	Hv+5NEPazlh29re5bM0w==	1-1-2
2	Howard	Roark	321qGEhqiVztNweFB4/Sgg==	1-1-3
3	Ayn	Rand	kZjnGbfVY1DGxZLxDCwujQ==	1-1-4

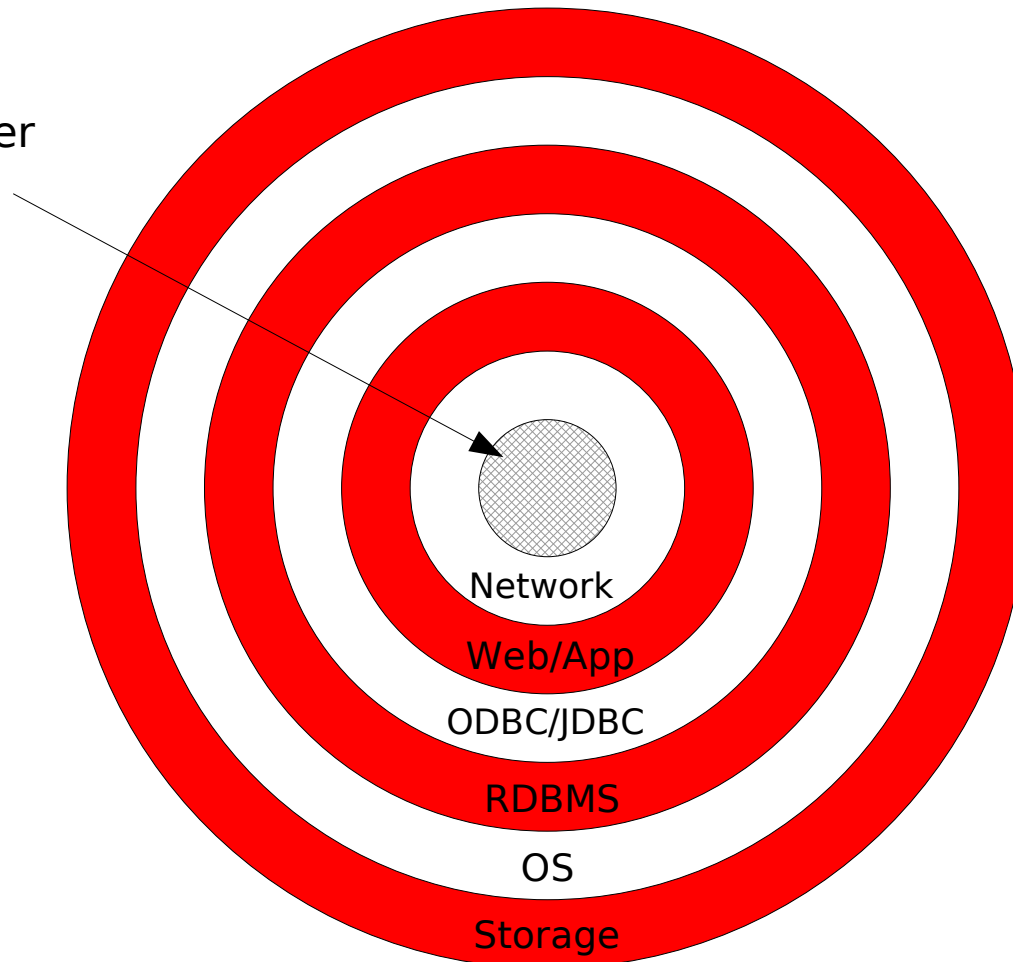
# Where should one encrypt?

- “Layer 7” encryption
  - Minimizes the attack surface
  - Focuses your mitigation efforts/dollars
  - Addresses the problem once and for all



# Another view of Layer 7

Application Layer

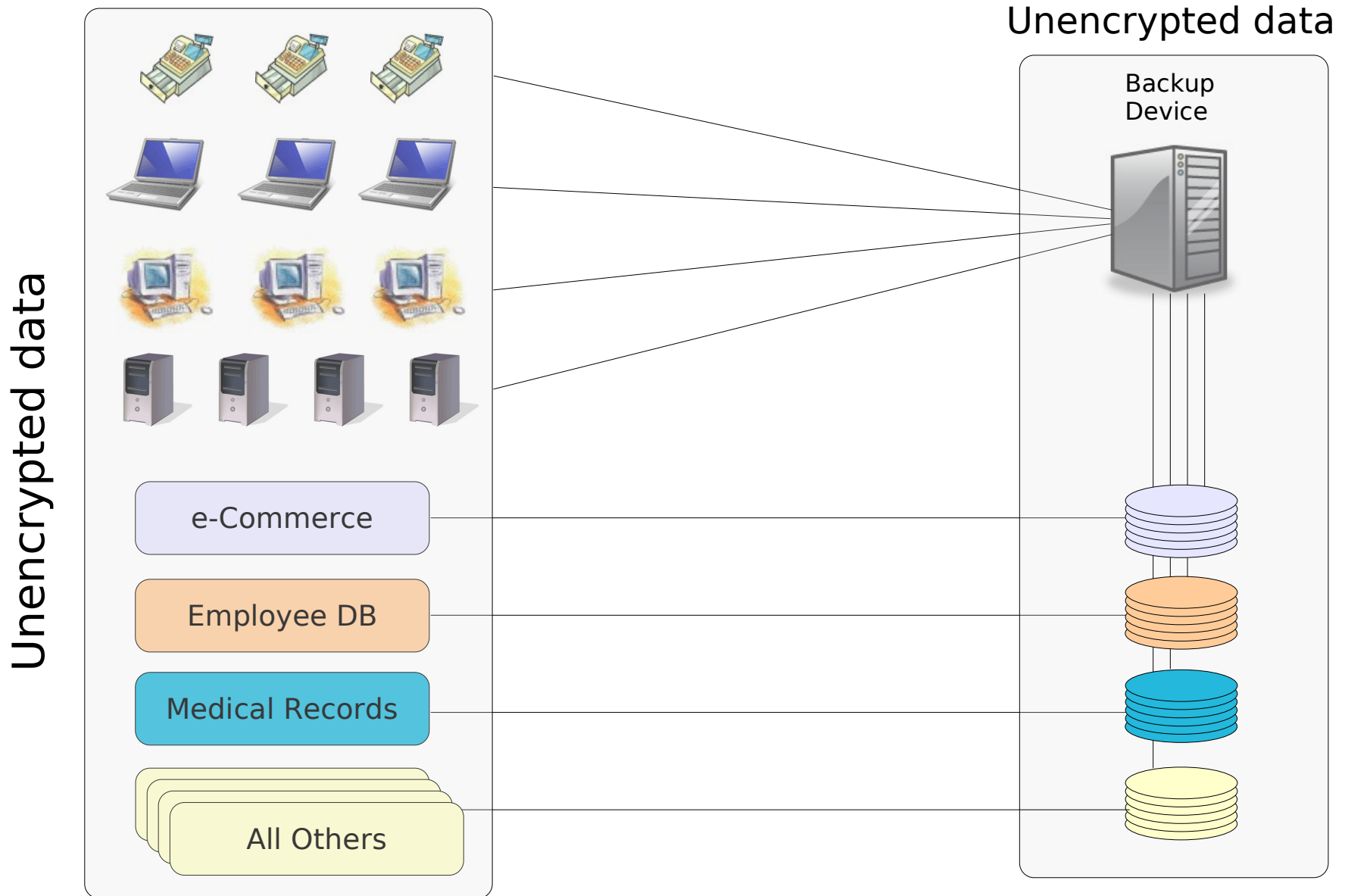


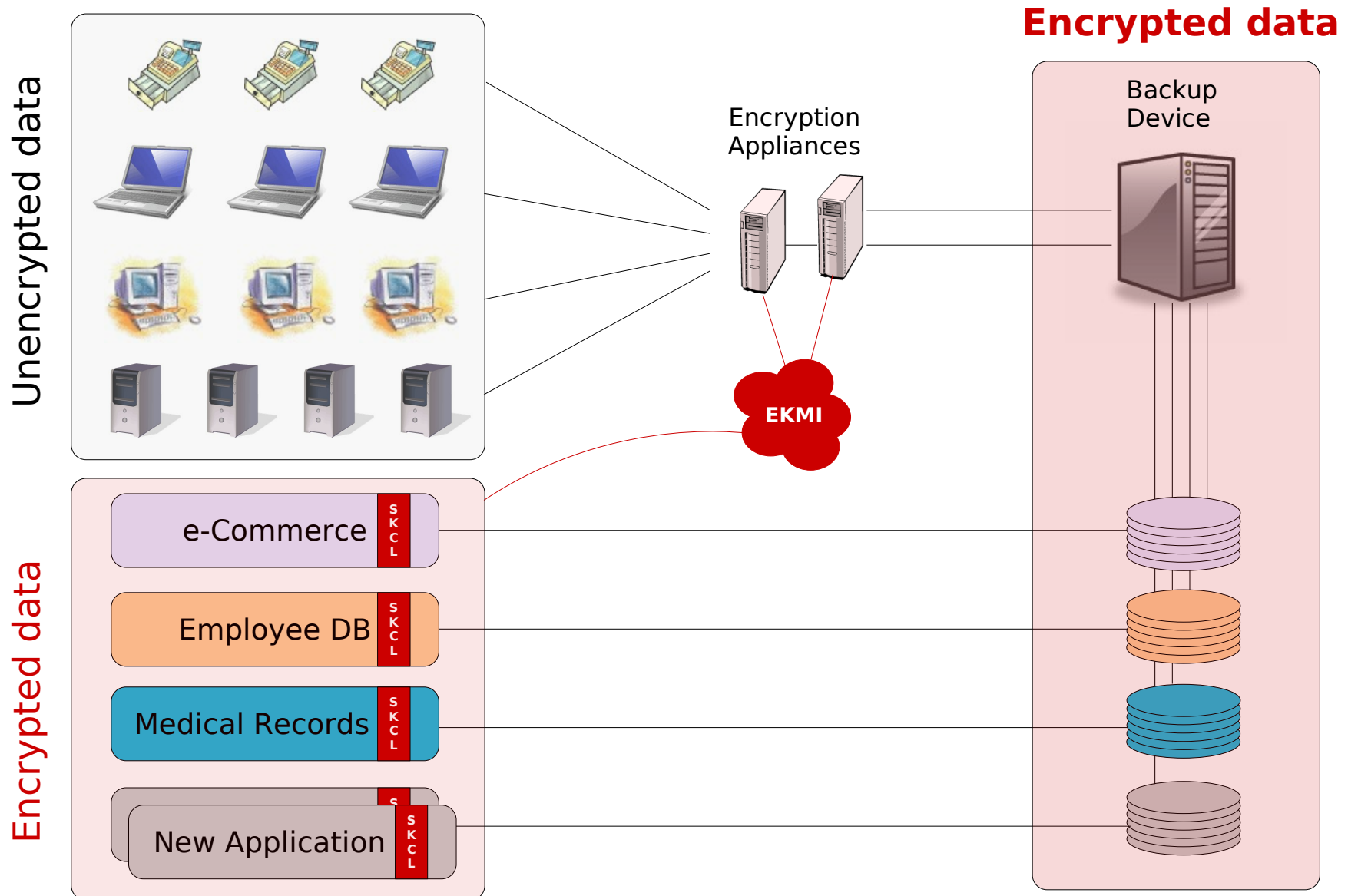
- Who is already encrypting at this layer?
  - PABP Application vendors
  - POS Application vendors
  - E-commerce platform vendors
  - Application Development teams who do not control the underlying IT infrastructure

- That doesn't cover all sensitive data in a company
- There are hundreds of applications
  - We'll never become secure if we have to modify every application
  - We'll never get the “buy-in” of the Business Units for wholesale revamping
- Solution:
  - An enterprise-encryption strategy in 3 phases

- Encrypt all backups
- Modify up to three (3) *legacy* applications with the most sensitive data in your enterprise, to encrypt **within** the application
- Encrypt data in all *new* applications **within** the applications
- Tell your software vendors you want them to encrypt data **within** the application in their new versions

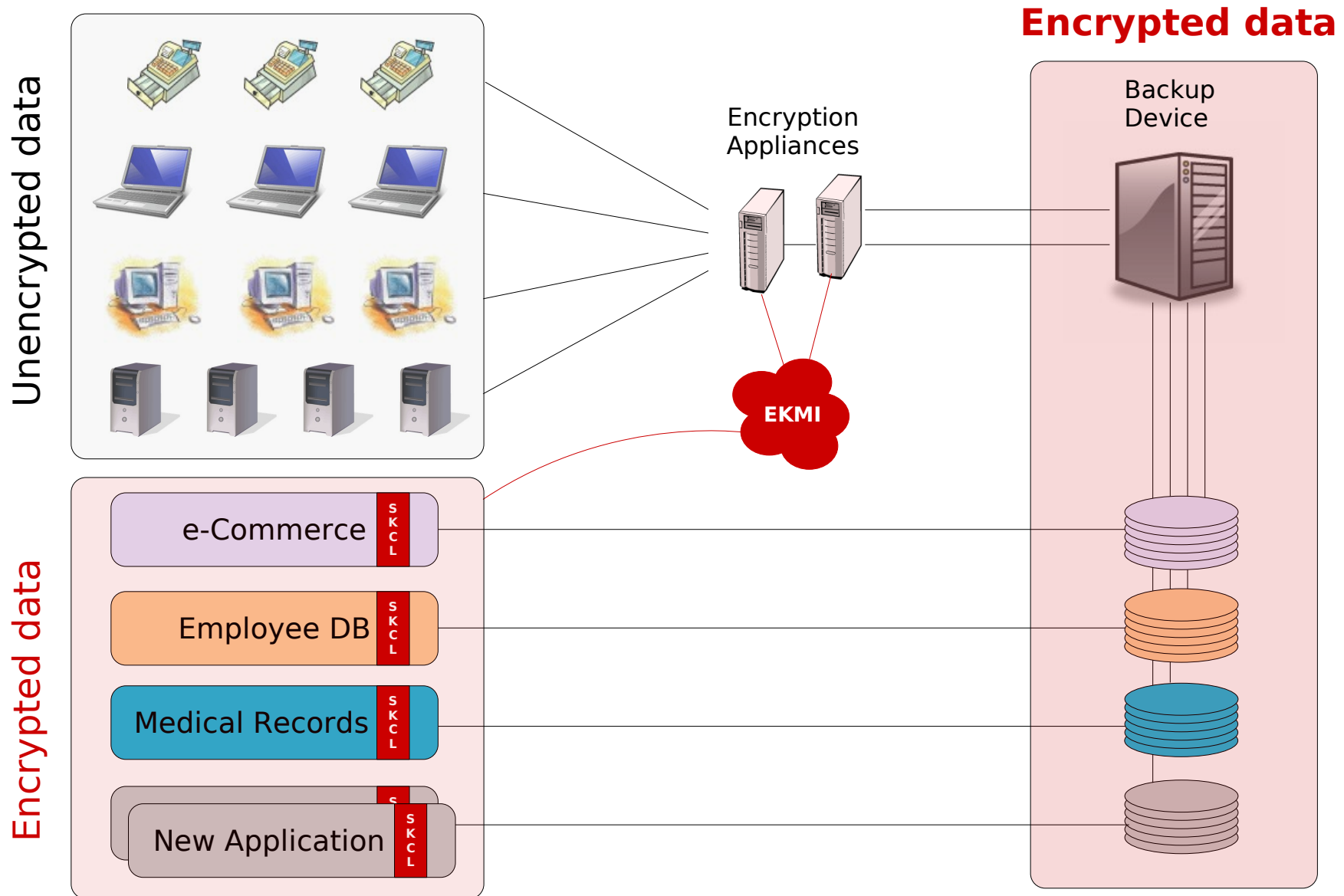
**BUT make sure all of the above use a standards-based KMS!**

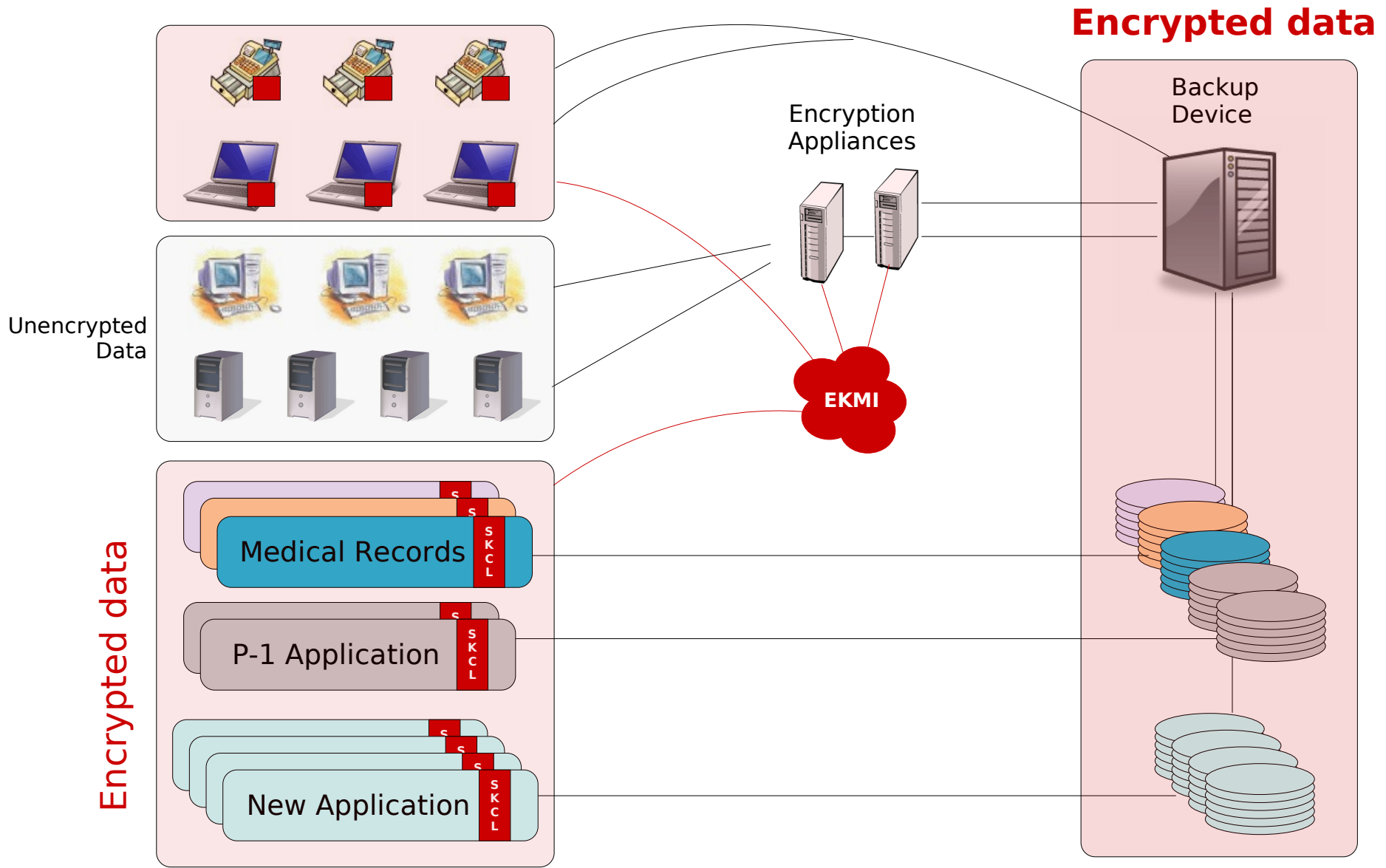




- Encrypt data on clients using **file-encryption**
  - Where necessary, continue to use an Encryption Appliance to encrypt backups
- Continue to modify legacy applications to encrypt data **within** the application
- Continue to encrypt data in new applications **within** the application
- Upgrade to new versions of software with data-encryption built into them

**Continue to use the standards-based KMS for the above.**





## Encrypted data

Unencrypted Data

Encrypted data

Encryption Appliances

EKMI

Backup Device

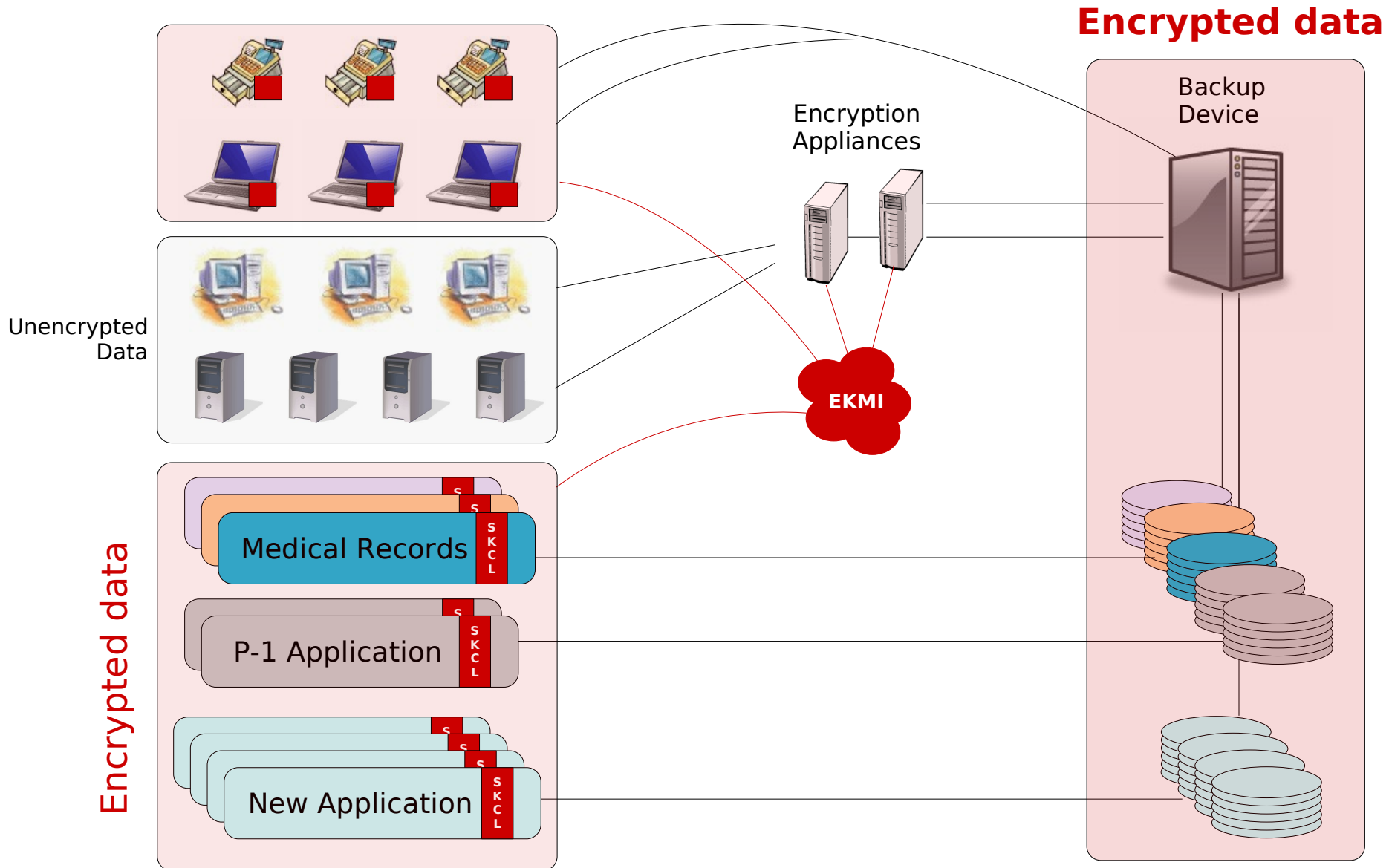
Medical Records

P-1 Application

New Application

- Encrypt data on clients using **application** encryption
- Continue to encrypt on clients using file-encryption, where necessary
- Continue to modify legacy applications
- Continue to encrypt data in all new applications
- Continue to upgrade to versions of software with data-encryption built into them

**Continue to use your standards-based KMS to manage keys across the enterprise!**



**Encrypted data**

Unencrypted Data

Encrypted data

Encryption Appliances

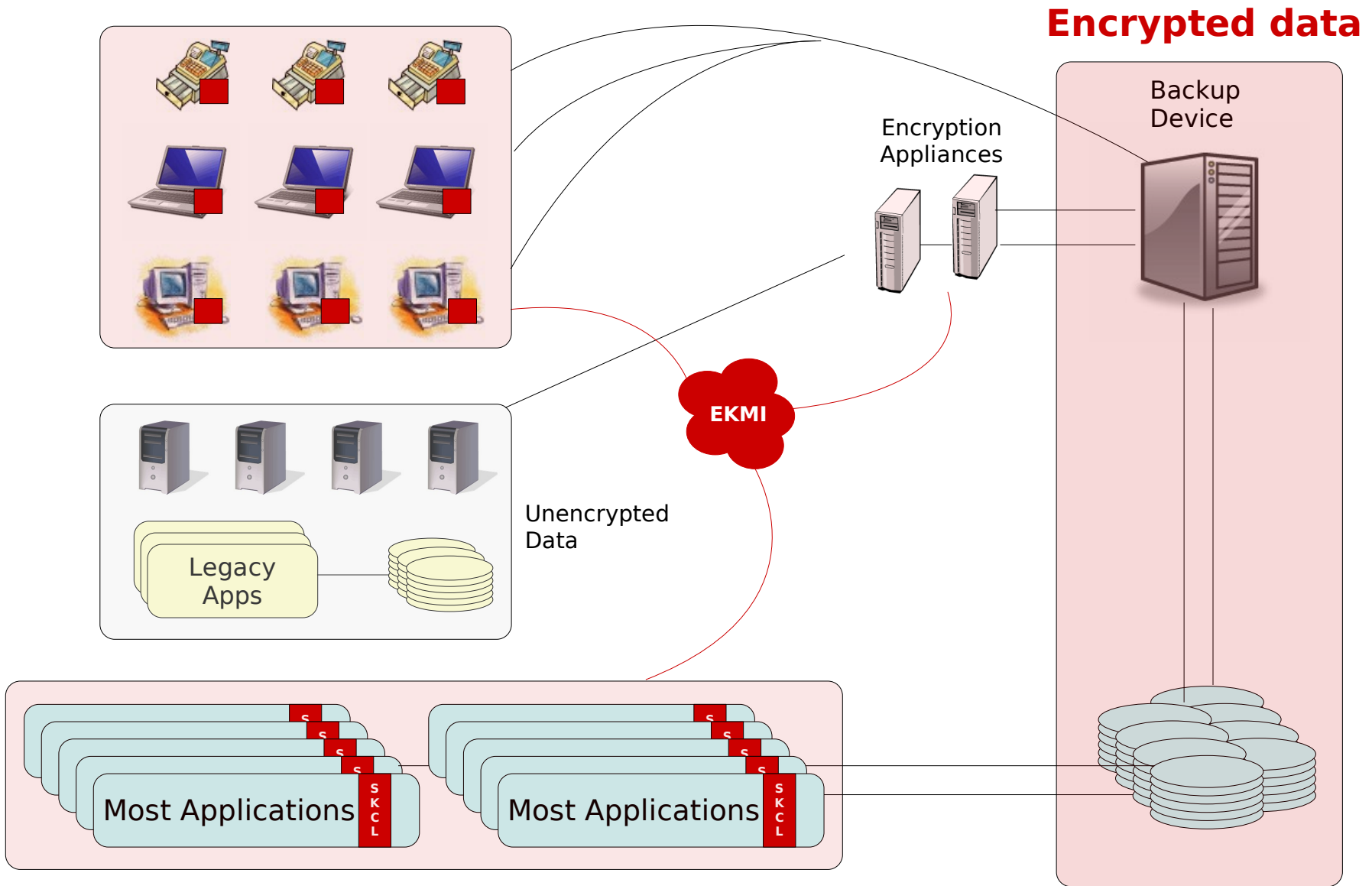
EKMI

Backup Device

Medical Records

P-1 Application

New Application



- SKMS will do for symmetric encryption key-management what DBMS did for data-management
  - Abstract cryptography out of the application
  - Abstract key-management out of the application
  - Become an “invisible” infrastructure component

- Open-source implementation available since Aug 2006
  - [www.strongkey.org](http://www.strongkey.org)
- 100% Java and supported on Linux, Windows, Solaris, OS/400 (client-only)
- C/C++ libraries available for Linux and Windows (Commercial license)
- Support forum
  - [www.strongauth.com/forum](http://www.strongauth.com/forum)

- Questions?
- Contact Information
  - [www.strongauth.com](http://www.strongauth.com)
  - [www.strongkey.org](http://www.strongkey.org)
  - [info@strongauth.com](mailto:info@strongauth.com)
  - (408) 331-2000