Java Database Access

Rick Cattell, JavaSoft

1. Relational database connectivity: JDBC

2. Higher-level relational APIs and tools

3. Object-oriented databases

4. Partners, summary

Overview

Demand for access to databases from Java

Increasing interest for general applications, not just Internet

Establish critical mass, libraries, tools for Java

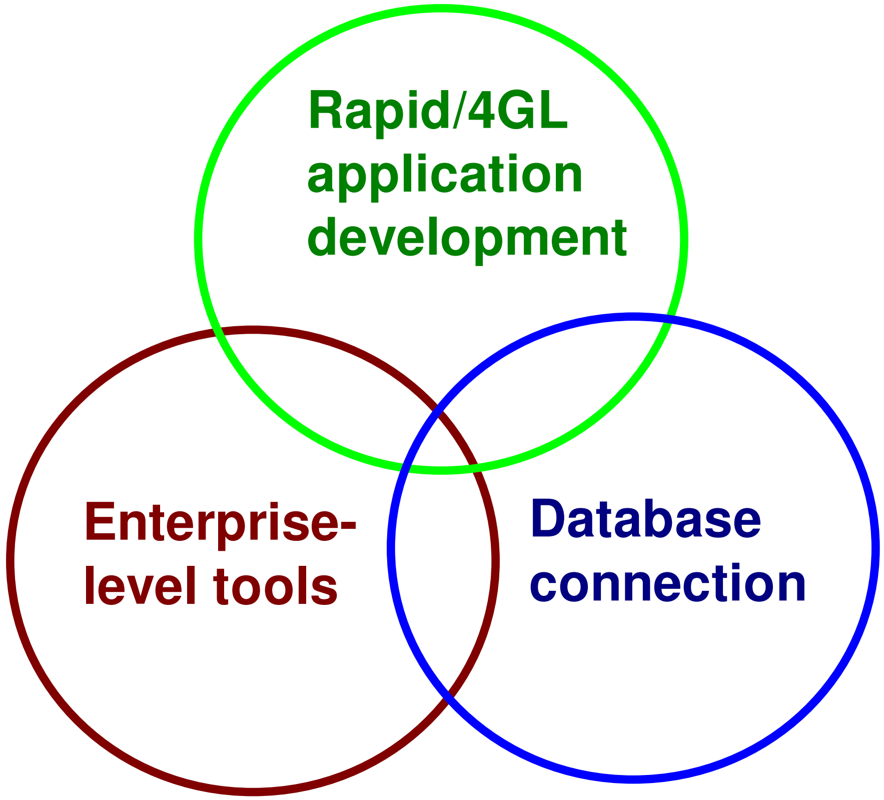
Have moved quickly with basic database connectivity

Not trying to "do it all ourselves": leverage partners, existing APIs

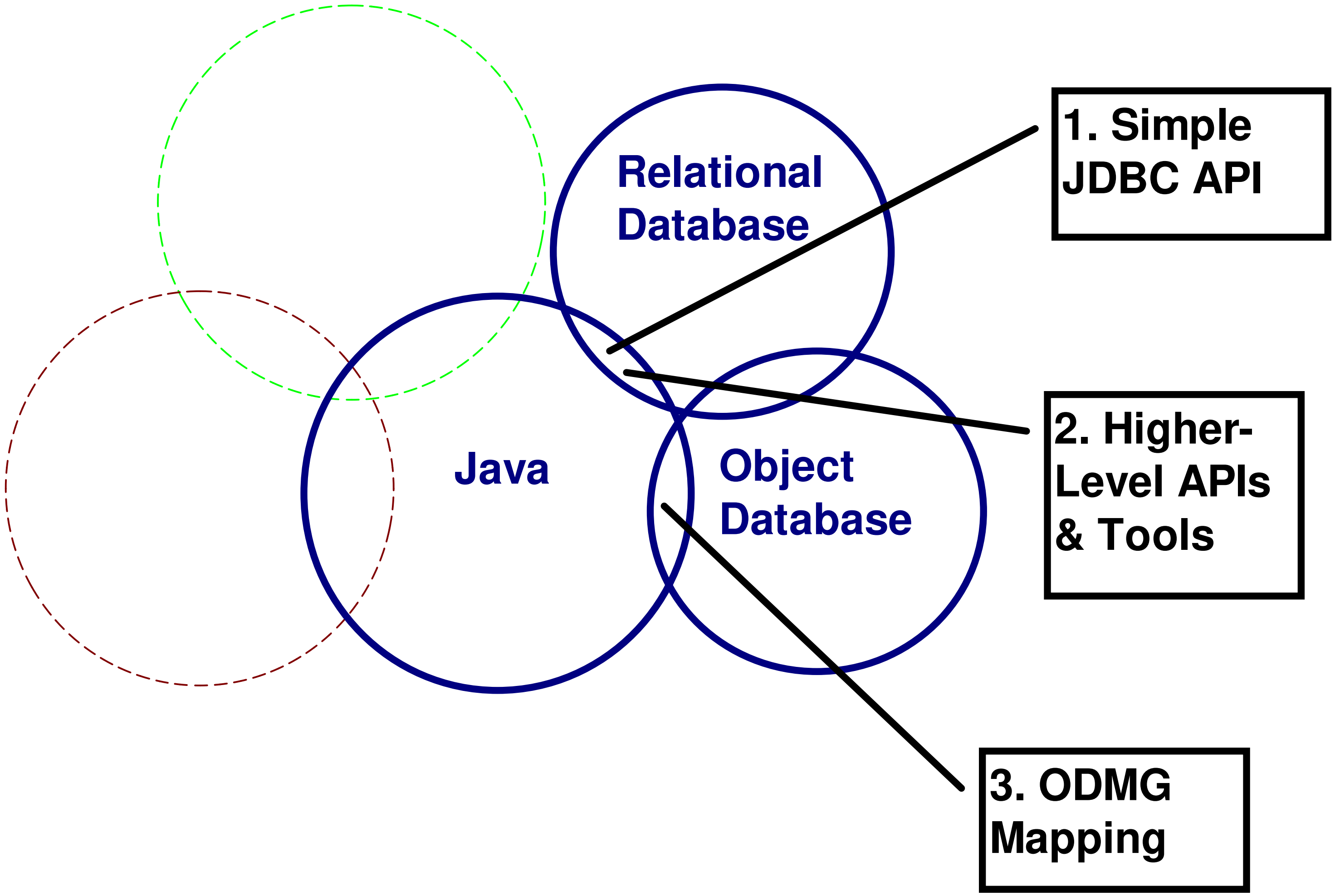
Why Java is Attractive for DB Applications

- Zero installation/administration
- Platform-independence
- Secure enterprise applications
- Internet connection
- Better language (than VB, C++, Smalltalk, COBOL)
- Increased productivity (no clobbers, leaks, void*...)
- However: the Java language alone is not enough

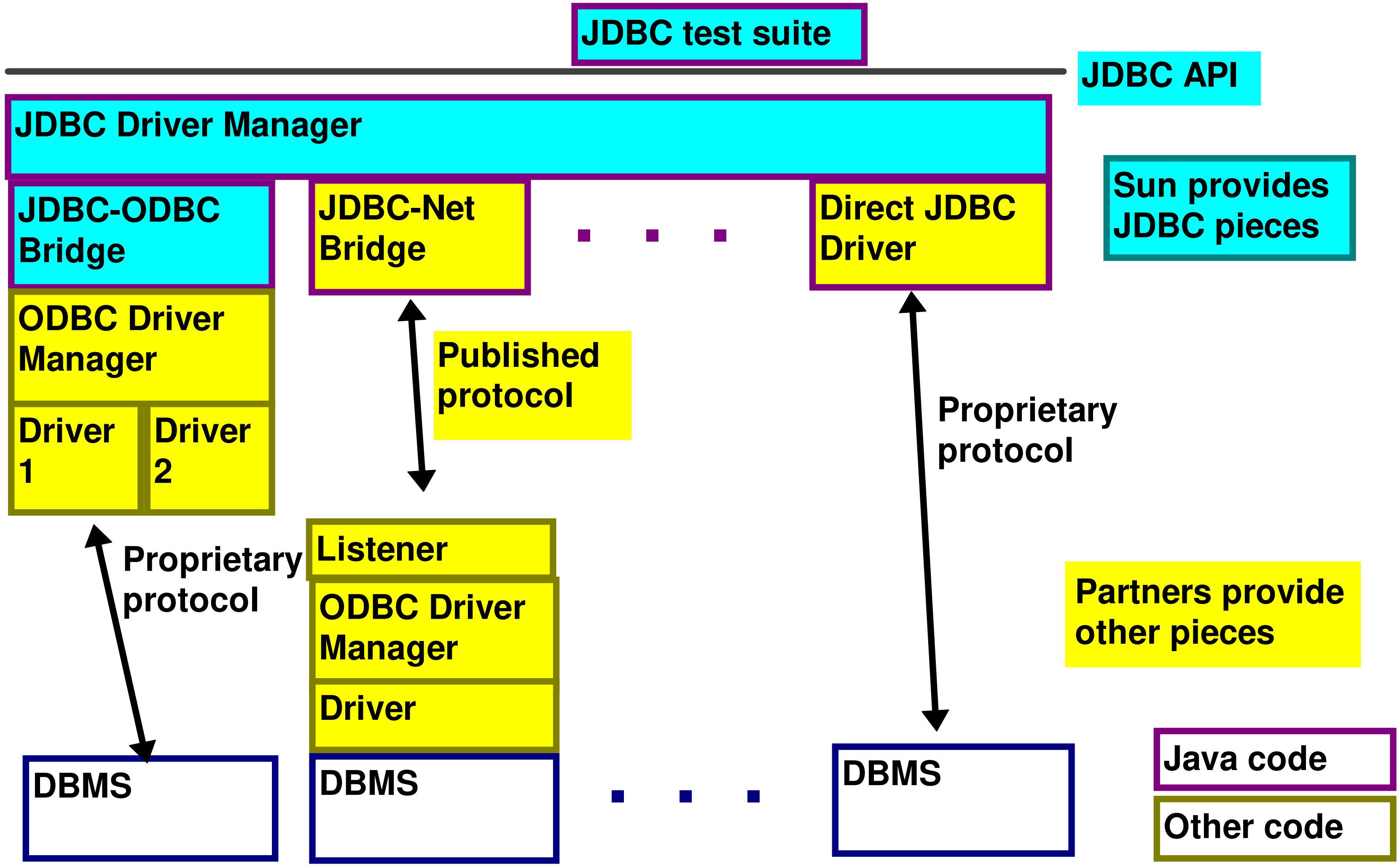
New Components for Enterprise Apps



Database Connectivity



1. SQL Database Interface: JDBC



JDBC Product

- Existing C database APIs not practical for Java
- Needed timely solution; easy to leverage ODBC
- Synergy for ISVs producing connectivity and tools
- Partnership with Intersolv and others
- See http://splash.javasoft.com/jdbc for more info

March	June	October
Alpha Spec Alpha DM Endorsees	Final spec, DM Drivers available Beta bridge, tests	More drivers available FCS ODBC bridge FCS test suite

JDBC Design

Generally patterned after ODBC and X/Open CLI

Faithful to Java with ease-of-use emphasis

Low-level API; build other APIs on top

Two kinds of users:

- **Programmers (need ease of use)**
- **Programs (need completeness, performance)**

Two modes of operation:

- **Untrusted applets and drivers on Internet**
- **Trusted code accessing company DBMS servers**

JDBC Features

Connection, Statement, ResultSet interfaces

PreparedStatement and CallableStatement for compilation and stored procedures

Metadata and dynamic access

Dynamic driver selection and loading

Database naming based on URLS; typically jdbc:<subprotocol>:<identifier>

JDBC Example

```
...
Connection conn = DriverManager.getConnection(
    “jdbc:odbc:sales”);
Statement stmt = conn.createStatement();
ResultSet rs = stmt.executeQuery(
    “SELECT Name, Sales FROM Customers”);
while (rs.next()) {
    String name = rs.getString(“Name”);
    int sales = rs.getInt(“Sales”);
    ...
```

2. Higher-Level Relational Tools and APIs

Embedded SQL for Java

Integration with application-building tools

Integration with 3-tier tools: RMI, CORBA

Enterprise repository and administration tools

Object/relational mapping

Object/Relational Mapping: Declarations

SQL:
```
CREATE TABLE CUSTOMER (
    CUSTID INTEGER NOT NULL,
    ADDRESS VARCHAR(50),
    SALESREP INTEGER,
    PRIMARY KEY (CUSTID),
    FOREIGN KEY (SALESREP) REFERENCES SALES);
```

Java:
```
class Customer {
    int CustID;
    String address;
    Sales salesRep }
```

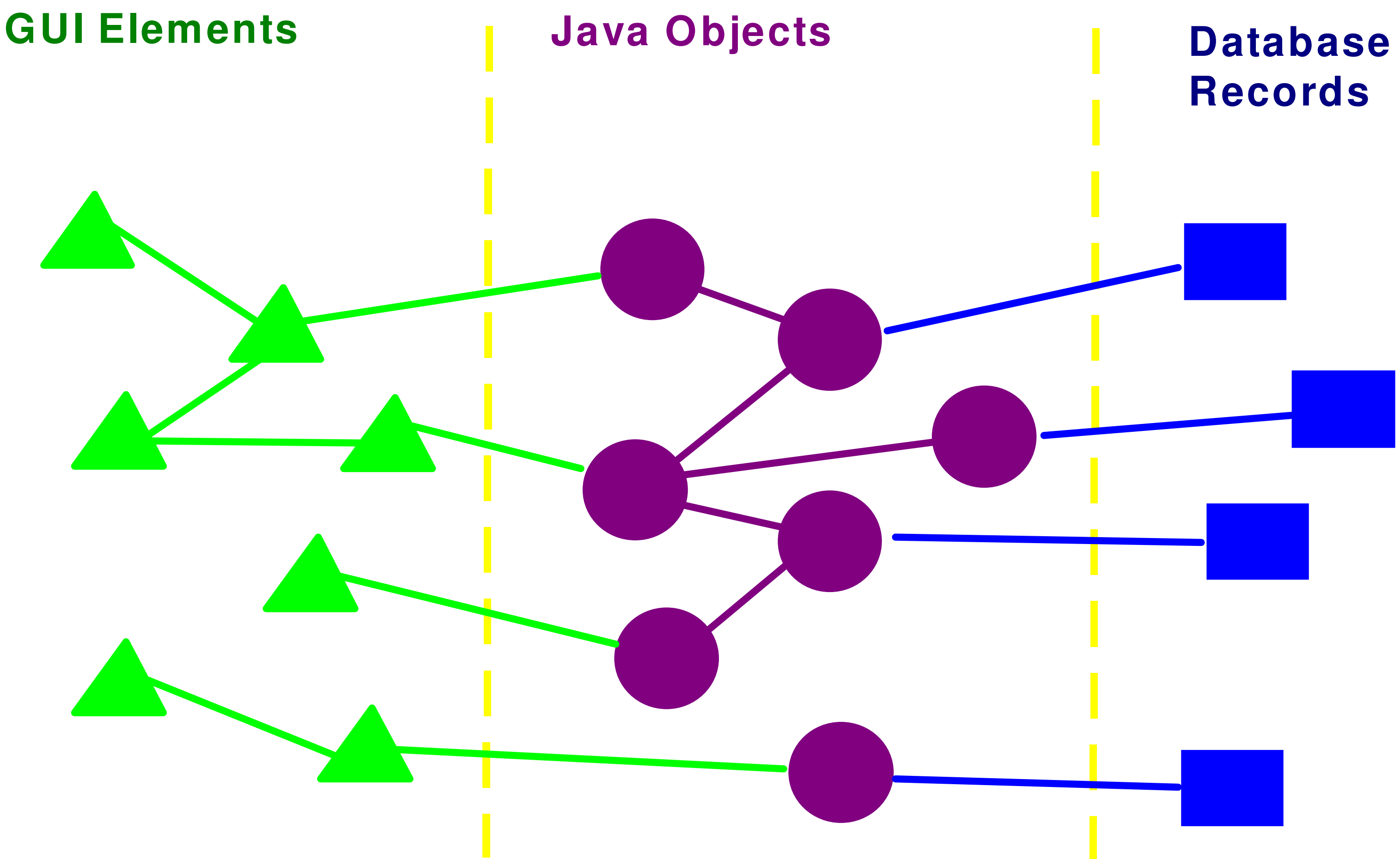
Automatically create Java class for each table in database schema (or vice versa), plus provide tools to embellish for many-to-one mappings, relationships, etc.

Object/Relational Mapping: Code

```
...
Transaction t = Transaction.create();
CustomerSet cs = CustomerSet.query(mycustomers);
Customer c = cs.select("CUSTID = 1234");
Sales s = c.salesRep;  /* follows foreign key */
c.address = newAddress;  /* obtains write lock */
s.sales = s.sales + thisOrder;
t.commit();  /* writes c and s back to database */
...
```

Java transaction object, database object, etc; transparently fetch/store Java objects from tables, map foreign keys to Java references

Object/Relational Mapping + GUI Tools



3. Object Databases for Java

Transparent persistence for Java objects

Java ideal for object databases since “safe”

Cached performance with more powerful data structures than relational

More transparent and faster than serialization

Not a JavaSoft product; working with ODMG consortium (Object Database Management Group): voting members GemStone, IBEX, O2, Object Design, Objectivity, POET, UniSQL, Versant

ODMG Timetable

Formed Java working group

Quickly reached agreement on initial specification

Several vendors well along on implementation

See http://www.odmg.org/ for more ODMG info

April	October	March '97
ODMG WG Draft spec	Review spec Java hook?	Freeze spec Initial products

ODMG Java Binding

ODMG class library: Database, transaction, query, and collection classes

Persistence orthogonal to type: Existing classes can have persistent and transient instances

Persistence by reachability: All objects reachable from designated named root database objects become persistent on transaction.commit

ODMG object model: Integrity constraints, can share data with persistent C++ and Smalltalk programs

Full database functionality: SQL2 superset OQL, transaction per thread, object locking

ODMG Example

```
Database.open("University", Database.ReadWrite);
Transaction t = new Transaction;
...
SetOfStudent mathematicians = Students.query(
    "exists s in this.takes: s.course.name = \"math\"");
Student joe = Students.select("id = 4132");
Professor oldAdvisor = joe.advisor;
joe.advisor = newAdvisor;
joe.address = "123 Main Street";
t.commit()
...
```

4. Java Database Products

Companies	Products this year
Bluestone	Sapphire/Web database app-builder
Borland	JDBC drivers and tools, DBMS, other Java development tools
BulletProof	JAGG JDBC/ODBC drivers and JDesignerPro database app-builder
IBM	DB2 JDBC driver, CGI scripting, app-building tools, Java stored procedures / user functions / triggers, Data Access Builder and Visual Age object/relational mapping on JDBC

Imaginary	**Postgres95 and mSQL JDBC drivers, mSQL DBMS**
Informix	**Database app-building tools, DBMS**
Intersolv	**SequeLink JDBC-net driver, JDBC drivers, ODBC drivers, joint development of JDBC/ODBC bridge and test suites**
O_2 Technology	**Object database, object/relational mapping**
Object Design	**Object database, object/relational mapping, JDBC driver**
Open Horizon	**Connection JDBC-net driver, security services, directory services, TP services**
OpenLink	**JDBC drivers**

Oracle	**Database application building tools, DBMS**
POET	**Object database**
SAS	**Java JDBC driver for Share*Net**
SCO	**SQL-Retriever JDBC/ODBC drivers**
Spider Software	**NetDynamics web/database application builder for JDBC and ODBC**
Sybase	**Optima++ Java database application development tools, DBMS**
Symantec	**JDBC drivers, Café app building tools and libraries**
Versant	**Object database**

Visigenic Software	**JDBC drivers, OpenChannel JDBC-net driver, ODBC drivers, source licensee partner**
WebLogic	**T3Server JDBC-net driver, dbKona API**
Working Set	**DataRamp JDBC-net driver and server**
XDB	**JetConnectPro JDBC/ODBC drivers, DB GUI classes on AWT, JetStream RDBMS/gateway**

For more partner info see JDBC web page

Note: Listed product names are trademarks of their respective companies; Java, JDBC, and JDBC-Compliant are trademarks.

Summary

JDBC: Quick to market, access legacy and relational database, direct use and generated code

Higher-level: Object-relational mapping, embedded SQL, integration with tools/repository

Object database: High-performance persistence for small footprint, embedded use

Partners: Not trying to do it all ourselves; focus on APIs so pieces plug together

Some Things I Wish I Learned in School

Research is not fun if people don't use your ideas or technology

Research may not be fun if someone else gets all the fame or fortune for your ideas

Nearly all startup companies fail: you should go there to have fun, not to get rich

Remember that most research and startup companies are *supposed* to fail: they are an experiment

Almost every research or development group tries to take on too much themselves

Corollary: Build on what's already successful (or *will* be successful)

“Technology transfer” is *very* hard: having good technology is only 20% of the solution

Corollary: Don’t worry too much about people stealing your good ideas — you’ll probably have to stuff these down their throats

The most popular technology is usually not the best

Learn basic principles of markets, management, and finance, not just technology; you must know and use these to avoid wasting years of your life (but learn from someone who has it right!)

There is little useful creative work in standards groups and they go on forever (also true of most conferences).

1% of the people in R&D are the key to most of the successes; these people have a rare combination of intelligence, drive, leadership, insight into technology futures, insight into the market, and other skills

Corollary: Think hard when hiring & funding projects

Don’t base a risky project on another risky one

Successful people frequently move as groups from one company to another

Venture capitalists pour a lot of money into experimental ideas even when they have a negative return on investment on average

Think hard about technology and market trends and base your work decisions on these... it’s worth the time

Be aware of software patents: you must play this game

Most managers are at heart opposed to telecommuting; it may never happen

It is worth taking time to increase your productivity

Prioritize tasks every day, every week, every year... don’t get stuck in an unproductive rut

Work as a team: build on other work, use existing infrastructure, delegate to others, leverage partners

Take a year off to have fun sometimes