

# Об архитектуре масштабируемых информационных сервисов

С.В. Знаменский

2 ноября 2012 г.

ИПС им. А.К. Айламазяна РАН

с. Веськово

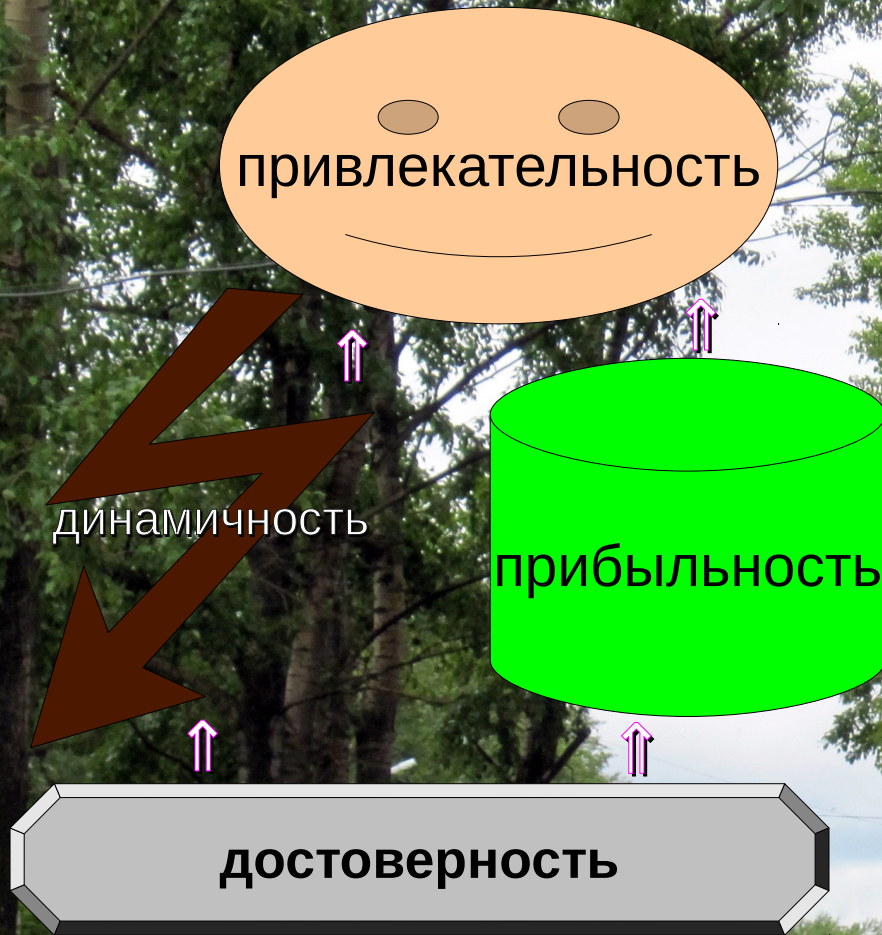
# Архитектура:

- The fundamental organization of a system embodied by
- its ***components***,
  - their ***relationships*** to each other and to the environment and
  - the ***principles*** guiding its ***design*** and ***evolution***.

*IEEE Standard for Architectural Description  
of Software-Intensive Systems  
(IEEE P1471/D5.3)*

# Фундаментальность понимается по-разному:

В логике учёных:



В логике политиков:



«Зри в корень!» *К. Прутков*

**Общая постановка задачи:**

*Информационный сервис от e-science*

**Гарантии достоверности данных (знаний, кода, ... ):**

«Dubito, ergo cogito, ergo sum»  
*René Descartes*

- Научная методология создания нового;
- Разносторонняя непрекращающаяся верификация  
( контроль качества)
- Полнота и точность сохранения исходных данных,  
результатов, методик (историчность)

**Table 43. Data Loss**

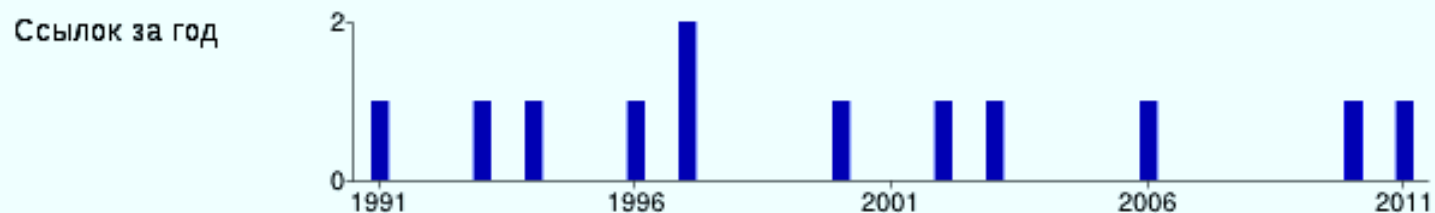
<b>Data Loss</b>	<b>Responses (1242)</b>	<b>Percentage (of 716 responders)</b>
Lack of funding	36	5%
Inadvertent human error	216	30%
Malicious hacking	6	1%
Mistakenly thought data not needed	49	7%
Equipment malfunction	173	24%
Lost media	73	10%
Mislabeled media	34	5%
Equipment obsolescence	76	11%
Software no longer recognizes data	88	12%
Physical disaster	29	4%
Data corruption	91	13%
I have not lost data	355	49%
Other	21	3%

**Table 63. Importance to make data available to future**

<b>Importance For Future</b>	<b>Responses</b>	<b>Percentage</b>
	<b>(764)</b>	
Very important	321	42%
Somewhat important	294	38%
Not very important	93	12%
Not important at all	31	4%
Not sure	11	1%
Other	14	2%

Название	<a href="#">A geometric criterion for strong linear convexity</a>
Авторы	Sergei Vital'evich Znamenskii
Дата публикации	1979/7/1
Название журнала	Functional Analysis and Its Applications
Том	13
Номер	3
Страницы	224-225
Издатель	Springer New York
Описание	We will call straight lines and planes affine submanifolds of $C^n$ of dimension $i$ and $n-i$ , respectively. For $z=(z_1, \dots, z_n) \in C^n$ , $w=(w_1, \dots, w_n) \in C^n$ we put $zw = z_1w_1 + \dots + z_nw_n$ . For any set $0 \neq E \subset C^n$ it is possible to identify the set $E = \{z \in C^n : zw = i \text{ for all } w \in E\}$ conjugate to it with the set of all planes which do not intersect $E$ . A set which is conjugate to a compact set is open, and a set which is conjugate to an open set is compact [1]. It is always true that $E \cap D \subset E$ . If $E = E$ , ie, if $C^n \setminus E$ is the union of a family of planes, then $E$ is said to be linearly convex in the ...

Всего ссылок [Цитируется: 12](#)



Статьи в Академии [A geometric criterion for strong linear convexity](#)  
 SV Znamenskii - Functional Analysis and Its Applications, 1979  
[Цитируется: 12](#) - [Похожие статьи](#) - [Все версии статьи \(4\)](#)

**Table 66. Preservation Priority Assessment**

	<b>Ability to Identify At-Risk Data (765)</b>		<b>Ability to Identify Important Data (765)</b>	
Very Easy	224	29%	249	33%
Somewhat easy	284	37%	303	40%
Somewhat difficult	122	16%	116	15%
Very difficult	27	4%	29	4%
Not sure	78	10%	39	5%
Other	30	4%	29	4%



**Table 32. Uniqueness of Data**

<b>Uniqueness</b>	<b>Responses (1725)</b>	<b>Percentage (of 747)</b>
I have observation data that is unique	338	45%
I have experimental data that is unique	370	50%
Data is unique due to the quantity and quality of the data	312	42%
Data is unique due to the level of uniformity and integration of the data	132	18%
Data is unique due to the longitudinal nature of the data	136	18%
Data is unique due to the added value of metadata	118	16%
Data is not unique and can be recreated from the original sources	113	15%
Data is unique due to the integration of unique analysis into the data	117	16%
Not sure how to describe the uniqueness of this data	107	14%
Other	51	7%

Stacy T. Kowalczyk.  
2011

**Table 31. Importance of Data Quality Control on Science**

<b>QC Importance</b>	<b>Responses (689)</b>	<b>Percentage</b>
Very important	434	63%
Somewhat important	169	25%
Not very important	45	7%
Not at all important	17	2%
Not sure	18	3%
Other	6	<1%

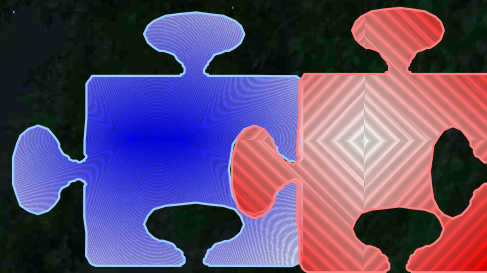
## *части архитектуры:*

### *Научные процессы (незаменимые)*

- ✦ Исследования и разработка*
- ✦ Управление ресурсами (сетевыми, вычислениями)*
- ✦ Мониторинг качества*

### *Заменяемые компоненты и модули*

- ✦ Оборудование*
- ✦ Алгоритмы*
- ✦ Библиотеки программ*
- ✦ Данные наблюдений*



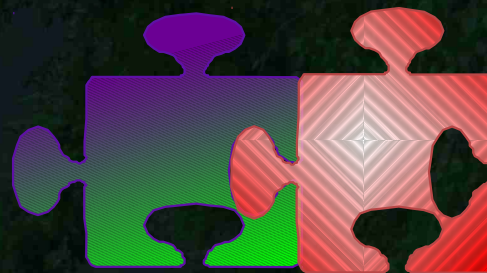
## части архитектуры:

### Научные процессы (незаменимые)

- ✦ Исследования и разработка
- ✦ Управление ресурсами (сетевыми, вычислениями)
- ✦ Мониторинг качества

### Заменяемые компоненты и модули

- ✦ Оборудование
- ✦ Алгоритмы
- ✦ Библиотеки программ
- ✦ Данные наблюдений



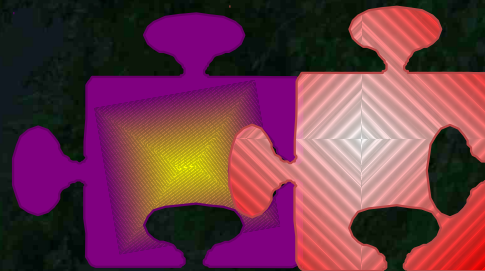
## части архитектуры:

*ISO9000-процессы* (и сущности с которыми они оперируют)

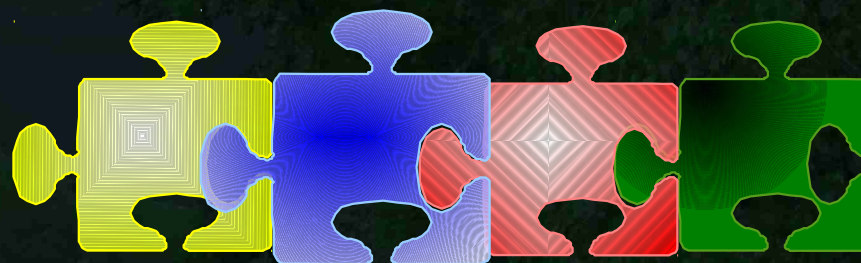
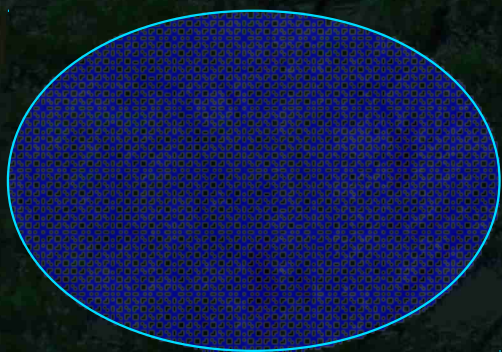
- ✦ *Исследования и разработка*
- ✦ *Управление ресурсами (сетевыми, вычислениями)*
- ✦ *Мониторинг качества*

*Заменяемые компоненты и модули*

- ✦ *Оборудование*
- ✦ *Алгоритмы*
- ✦ *Библиотеки программ*
- ✦ *Данные наблюдений*

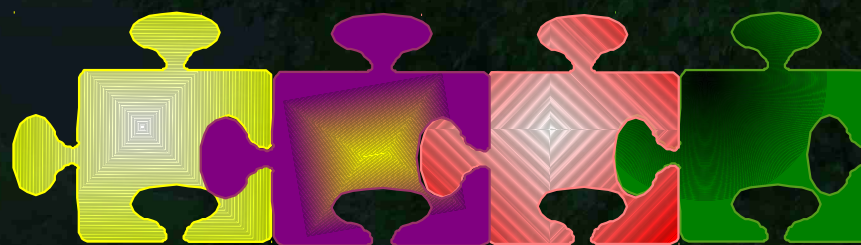
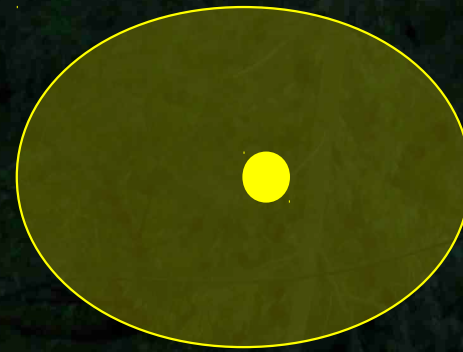
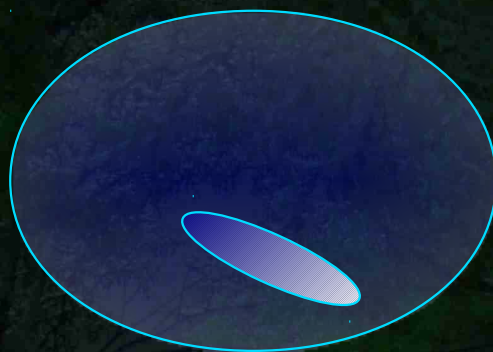
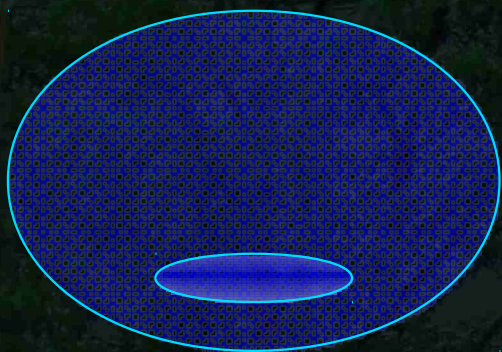


# Сущности



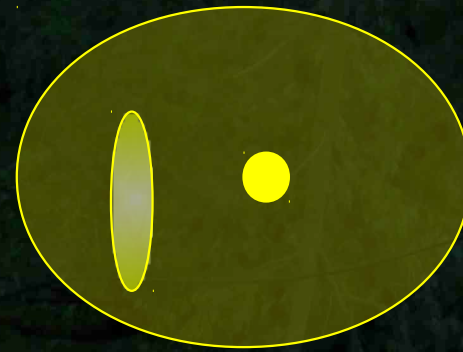
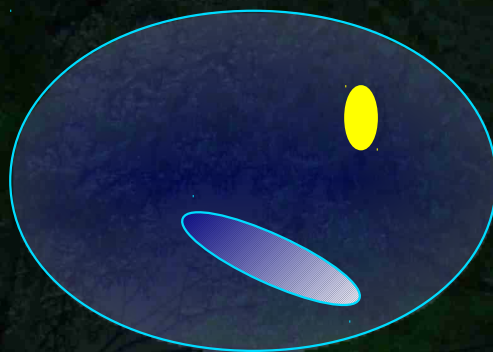
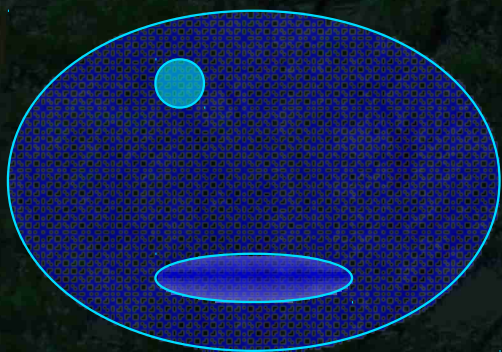
# Объекты

# Сущности



# Объекты

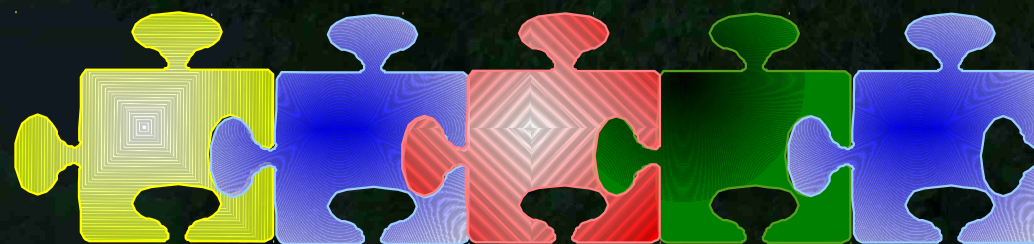
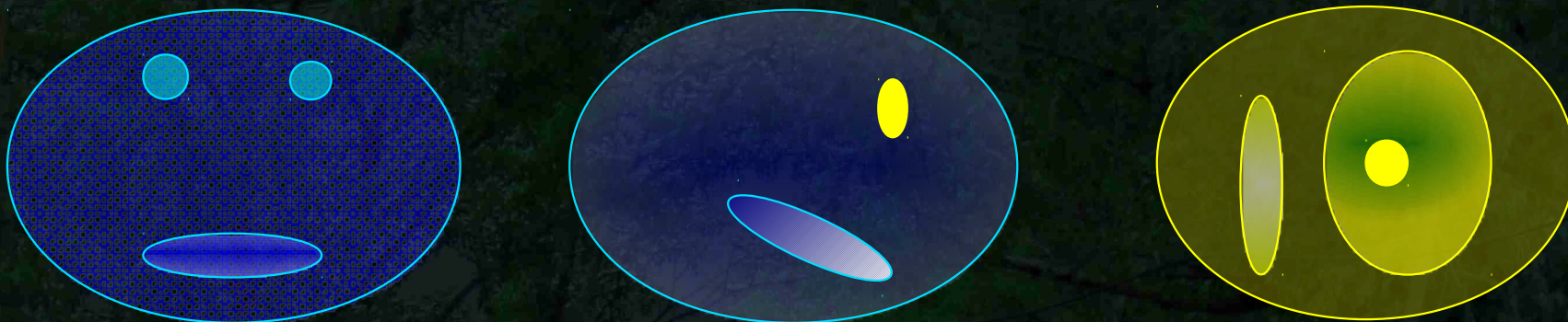
# Эволюционирующая иерархия сущностей



Объекты

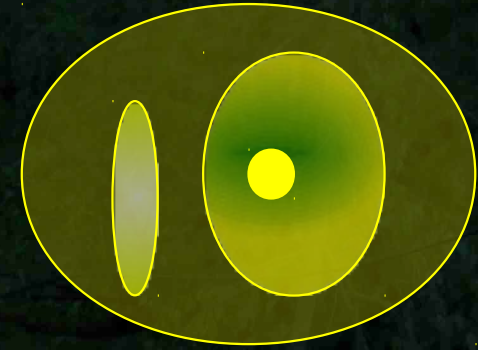
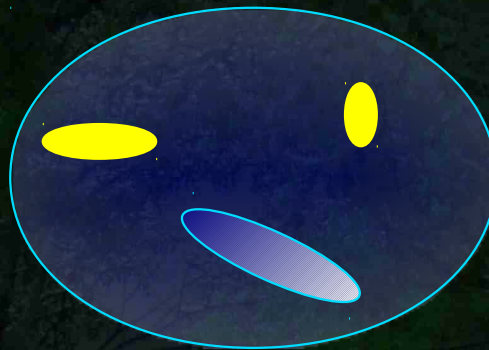
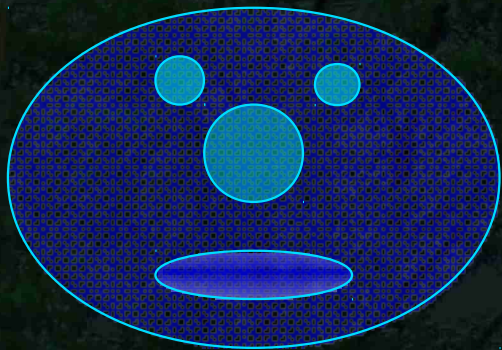


# Эволюционирующая иерархия сущностей

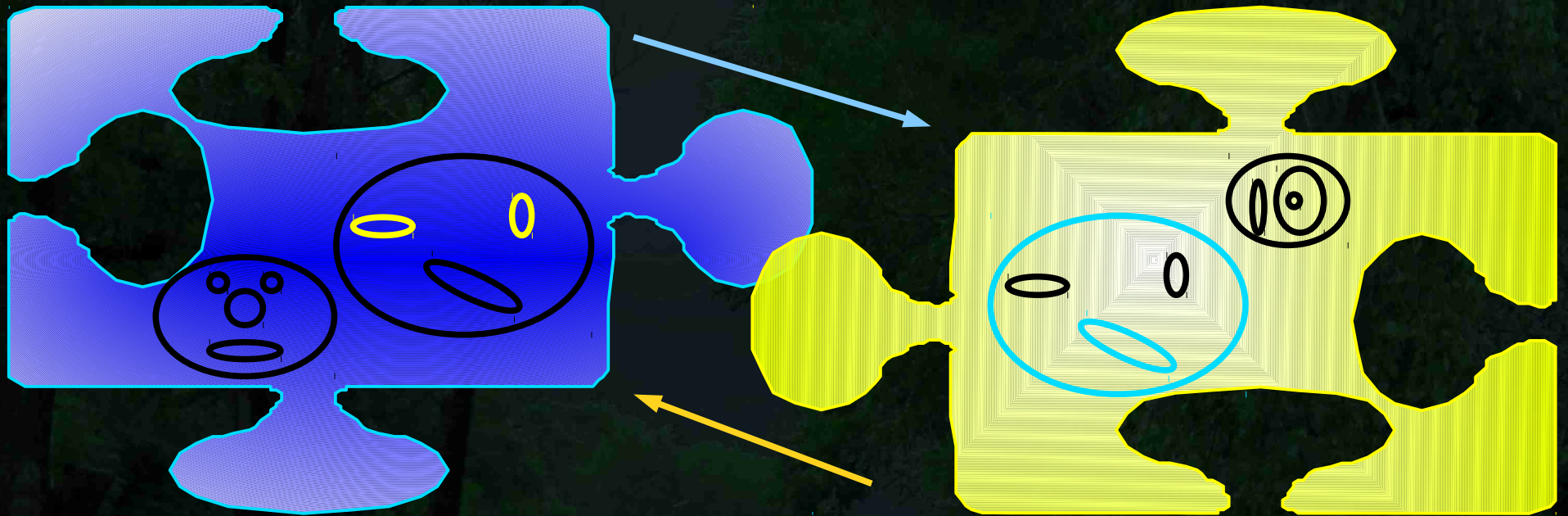


калейдоскоп сменяющихся объектов

# Эволюционирующая иерархия сущностей

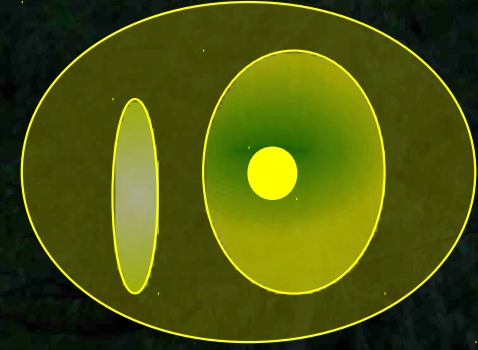
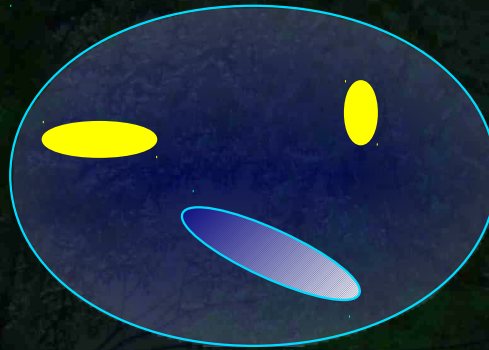
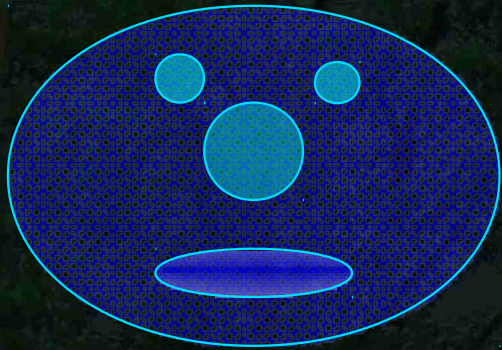


?  ?  ?  ? Основа архитектуры ?  ?  ?  ?

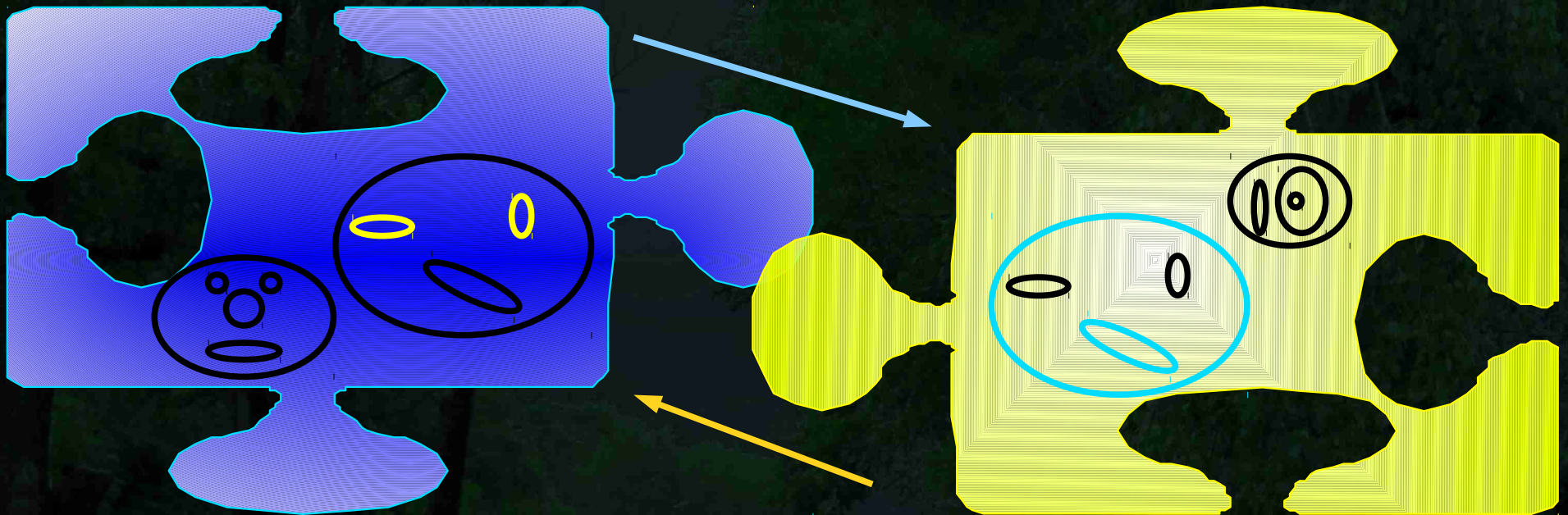


калейдоскоп сменяющихся объектов

# Глобальное адресное пространство иерархии метаданных



Разделяемая память на основе сети ретроспективных СУБД



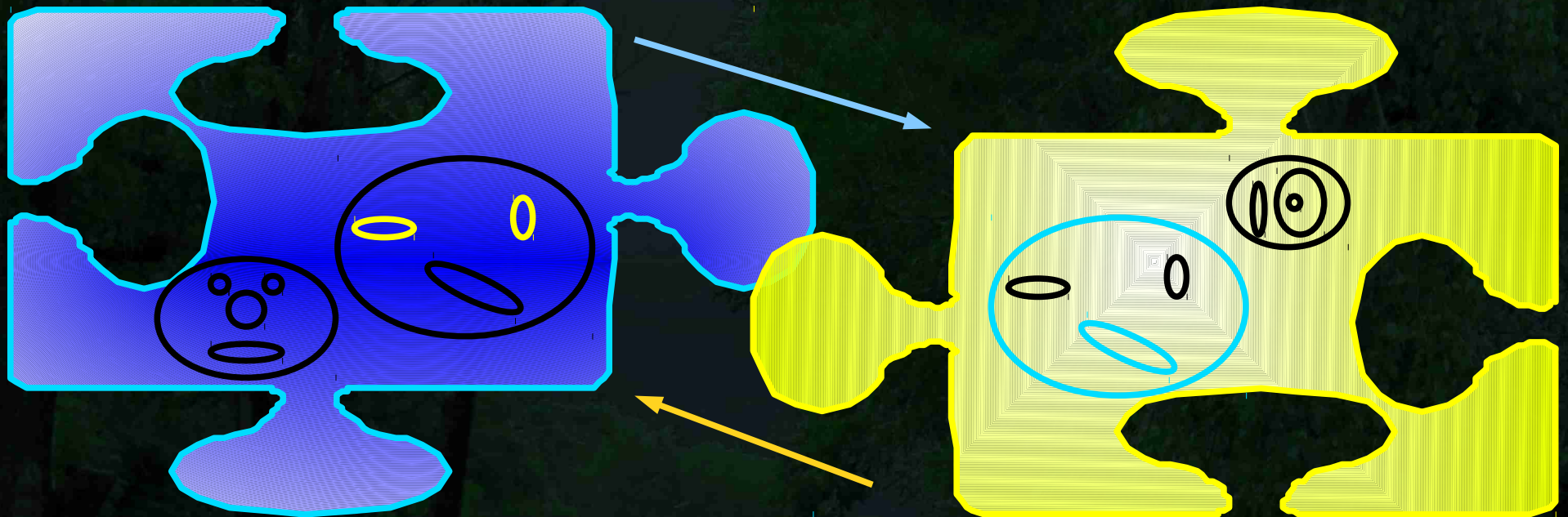
хранит уникальные данные с неподдельной историей

# Глобальное адресное пространство иерархии метаданных

освобождает прикладного программиста от сложностей обеспечения

- сохранности данных и восстановления при поломках,
- исправления ошибок в информации,
- доступности неискажённой истории,
- согласованности распределённой информации,
- управления доступом к данным и приоритетами.

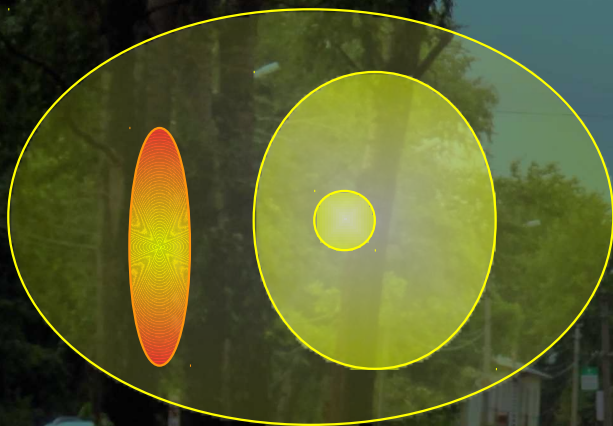
Разделяемая память на основе сети ретроспективных СУБД



хранит уникальные данные с неподдельной историей

# Ретроспективная СУБД

- ✦ Быстро запоминает объект с пометкой времени сохранения;
- ✦ Быстро возвращает актуальную на заданный момент версию и её время;
- ✦ Пересылает подписчикам изменения в интересных данных;
- ✦ Высвобождает место за счёт разумной деградации истории;
- ✦ Показывает наличие изменений, пригодных к обработке;
- ✦ Поддерживает ветвление версий символическими ссылками:



# Ретроспективная СУБД

- ✦ Быстро запоминает объект с пометкой времени сохранения;
- ✦ Быстро возвращает актуальную на заданный момент версию и её время;
- ✦ Пересылает подписчикам изменения в интересных данных;
- ✦ Высвобождает место за счёт разумной деградации истории;
- ✦ Показывает наличие изменений, пригодных к обработке;
- ✦ Поддерживает ветвление версий символическими ссылками:



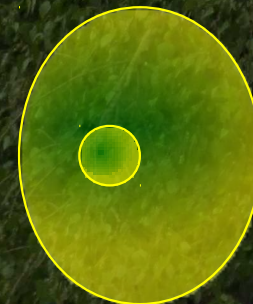
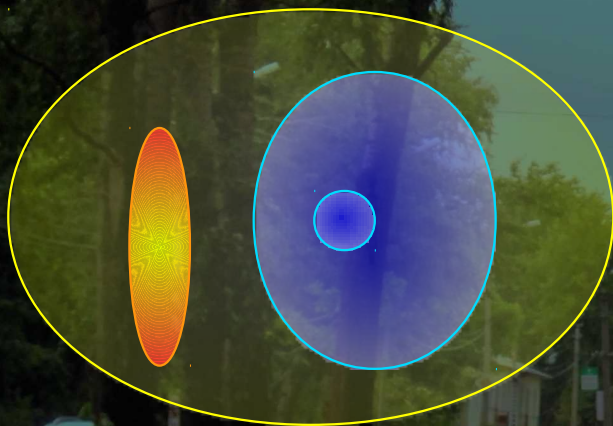
# Ретроспективная СУБД

- ✦ Быстро запоминает объект с пометкой времени сохранения;
- ✦ Быстро возвращает актуальную на заданный момент версию и её время;
- ✦ Пересылает подписчикам изменения в интересных данных;
- ✦ Высвобождает место за счёт разумной деградации истории;
- ✦ Показывает наличие изменений, пригодных к обработке;
- ✦ Поддерживает ветвление версий символическими ссылками:



# Ретроспективная СУБД

- ✦ Быстро запоминает объект с пометкой времени сохранения;
- ✦ Быстро возвращает актуальную на заданный момент версию и её время;
- ✦ Пересылает подписчикам изменения в интересных данных;
- ✦ Высвобождает место за счёт разумной деградации истории;
- ✦ Показывает наличие изменений, пригодных к обработке;
- ✦ Поддерживает ветвление версий символическими ссылками:





# Ретроспективная СУБД

Хранит данные в B+tree с ключами вида  
конкатенация идентификатора данного и строки времени.

В конце строки времени всегда стоит момент записи.  
В начале строки времени стоит момент актуальности.

Актуальная на заданный момент версия получается за  
логарифмическое время.

Наличие правки, время правки и предыдущая версия доступны  
прикладному программисту.

Проблема бабочки: исправления старых данных дорогостоящи.

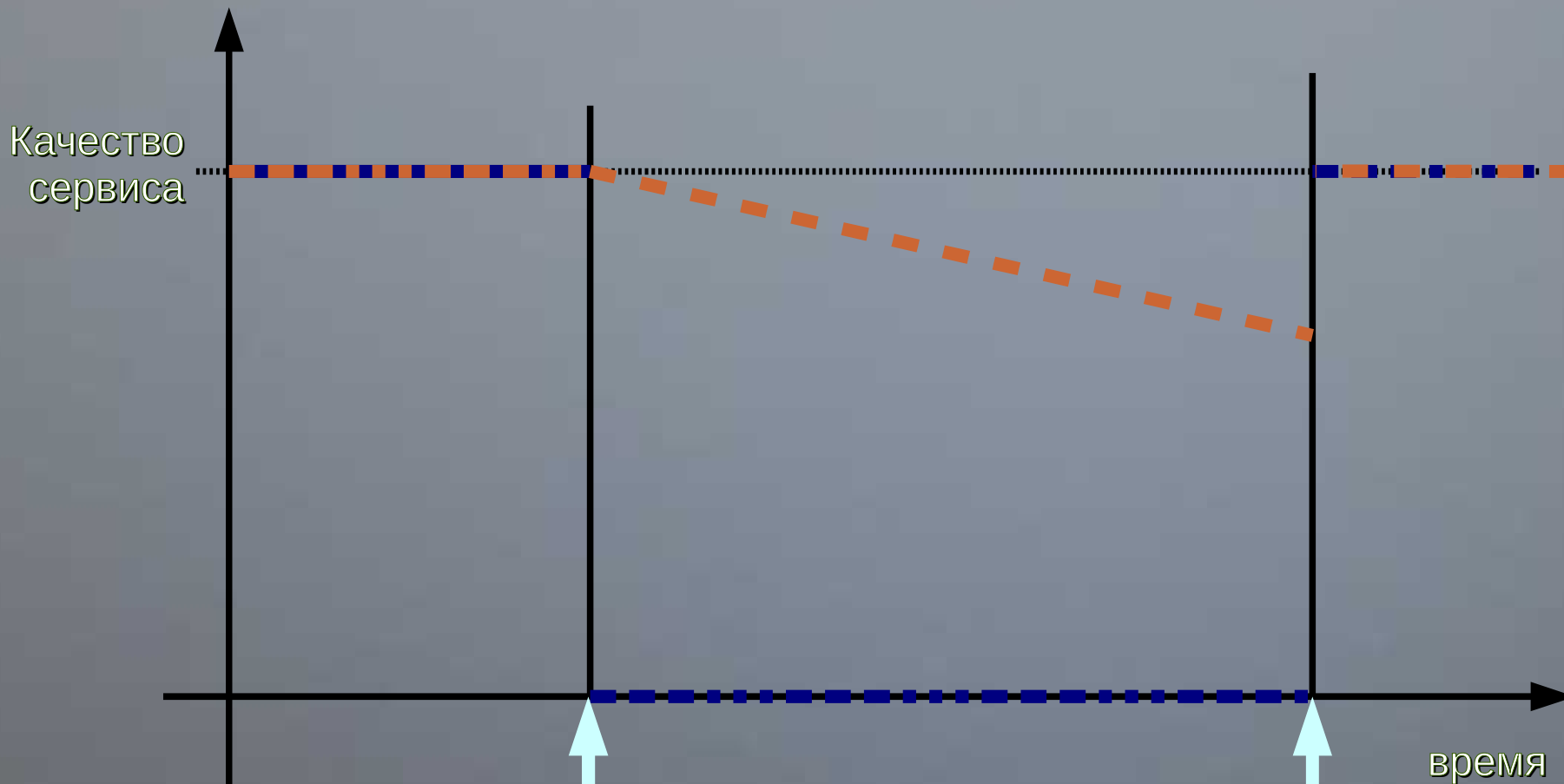
# Ретроспективная парадигма!

☠ Текущее состояние недоступно!

⚠ Запросы указывают контекст и время  $t$ .

⚡ Ответы заранее ритмично готовятся.

⇒ Немедленный ответ точен на известный момент  $< t$ .



- Система реального времени
- Ретроспективная система

Временная частичная неработоспособность, из-за обновления, ремонта, задержек связи или перестройки

Система восстановлена

# Ретроспективная парадигма!

☠ Текущее состояние недоступно!

⚠ Запросы указывают контекст и время  $t$ .

⚡ Ответы заранее ритмично готовятся.

⇒ Немедленный ответ точен на известный момент  $< t$ .

Путь к высоко отзывчивой надёжной долгоживущей системе?