

HISPASEC

**Praktyczne podejście do
testowania bezpieczeństwa
implementacji obsługi
formatów danych**

by Gynvael Coldwind

Dramatis Personæ

Gynvael Coldwind

- obecnie spec. ds. bezp. IT @ Hispasec
- wcześniej ArcaBit
- autor kilku artykułów (Hakin9 i Xploit)
- prowadzi bloga technicznego (<http://gynvael.coldwind.pl>)
- team Vexillium (<http://vexillium.org>)

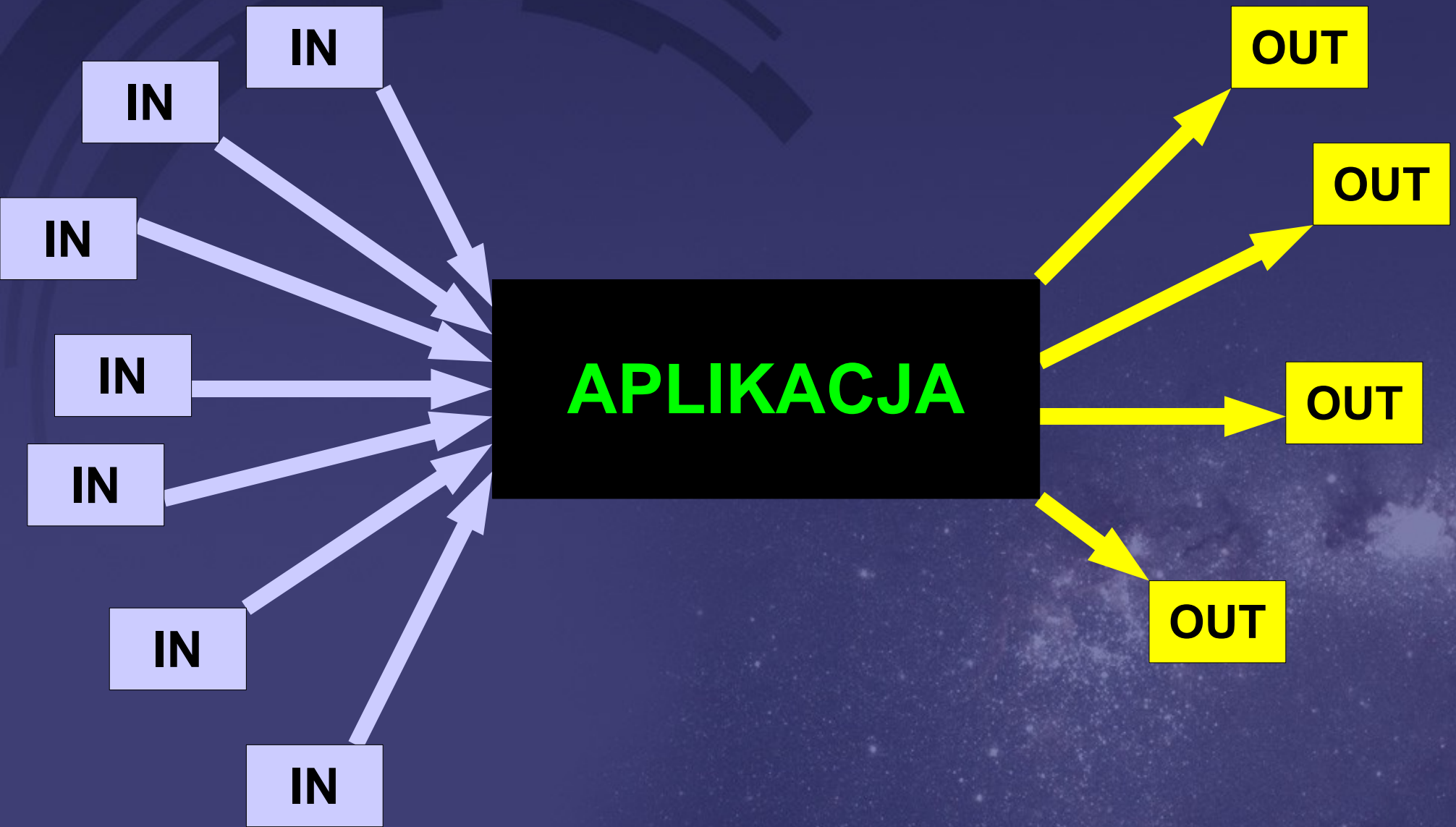


Menu

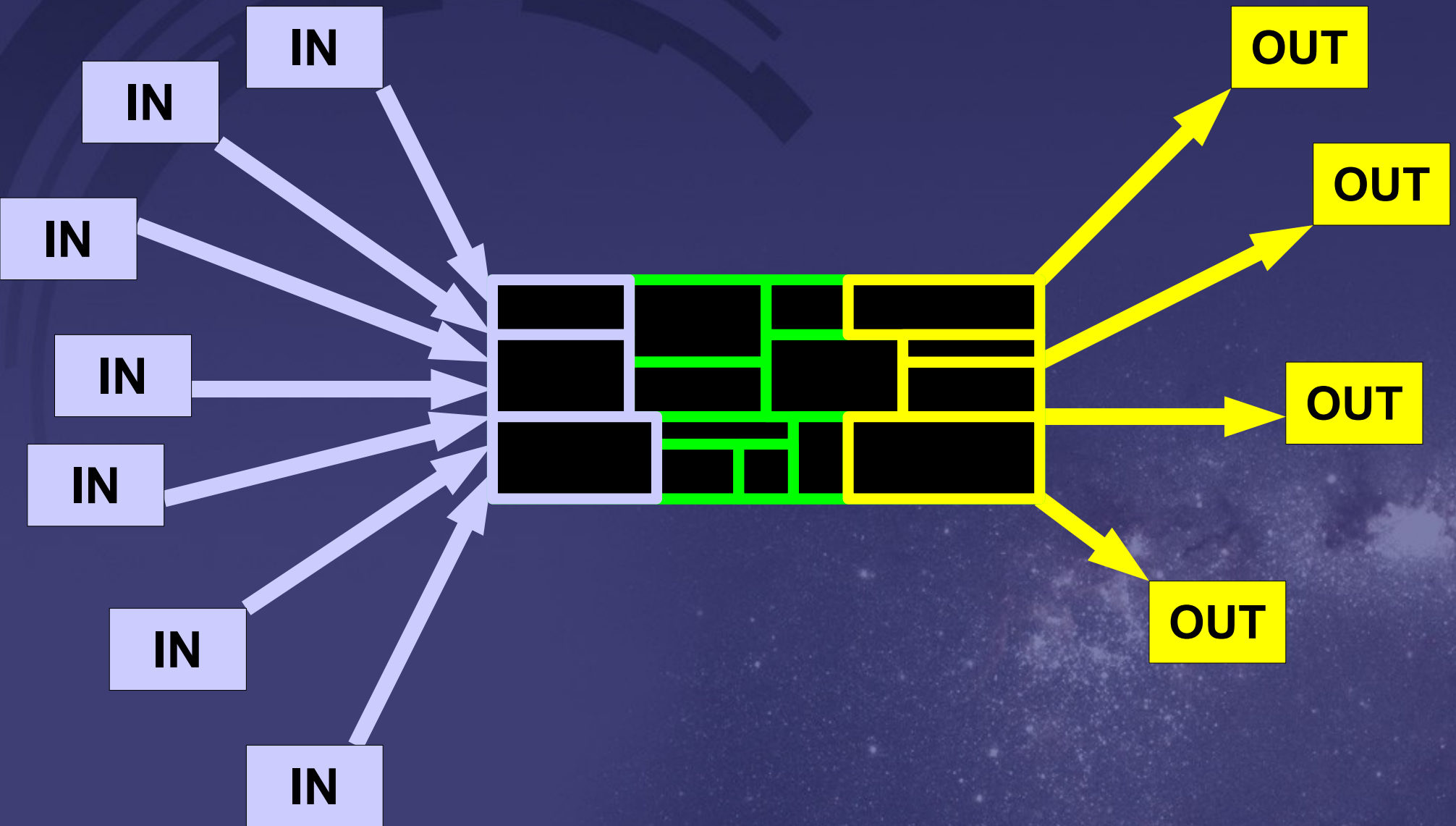
1. Budowa aplikacji a wektory ataku
2. Jak szukać błędów?
3. Różnice w interpretacji dokumentacji
4. Case study – BMP
5. Case study – GIF
6. Inne – RAR, ZIP, FTP
7. Podsumowanie



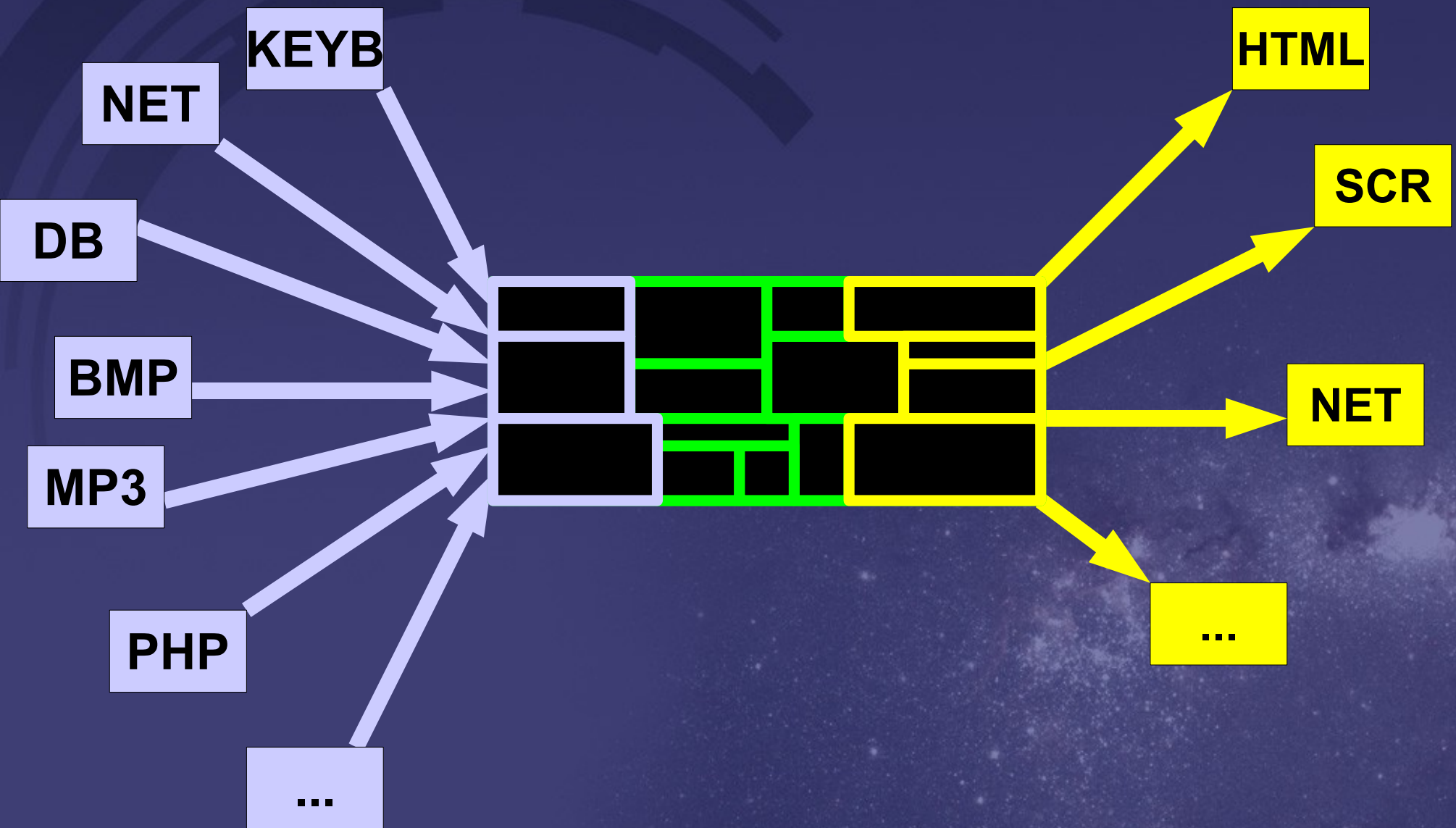
Budowa aplikacji



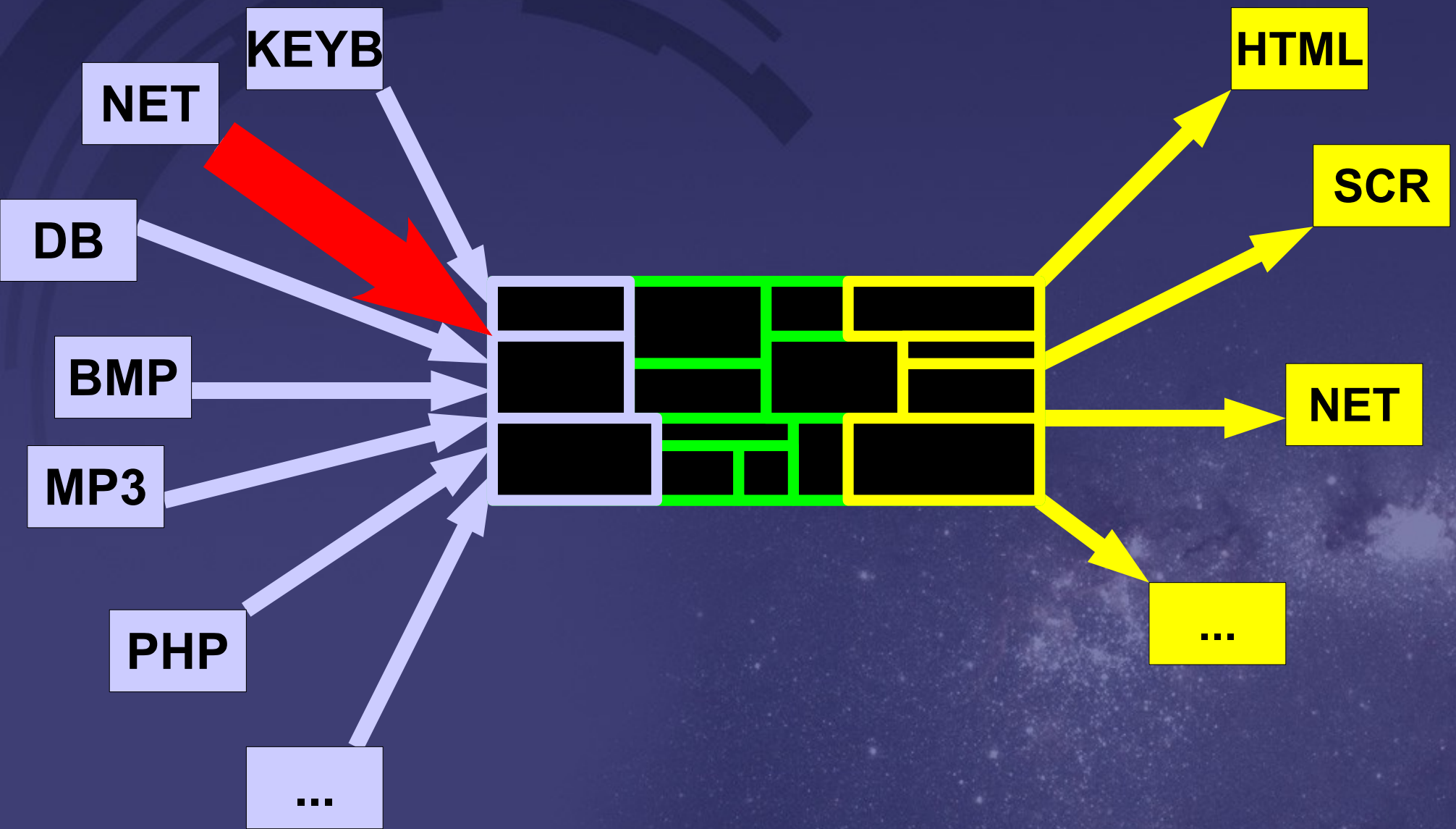
Aplikacja składa się z wielu modułów



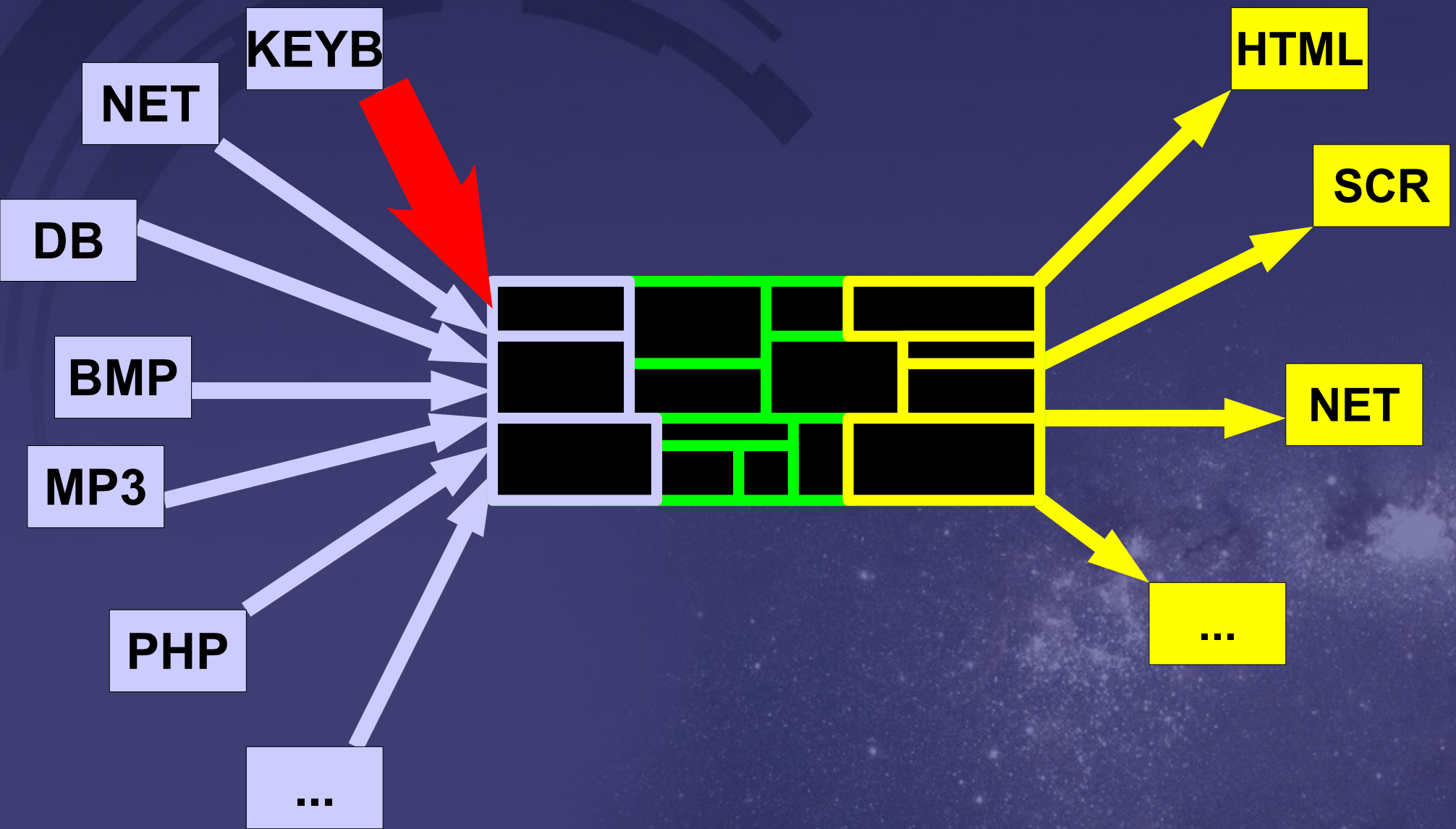
Wejście, przetwarzanie, wyjście



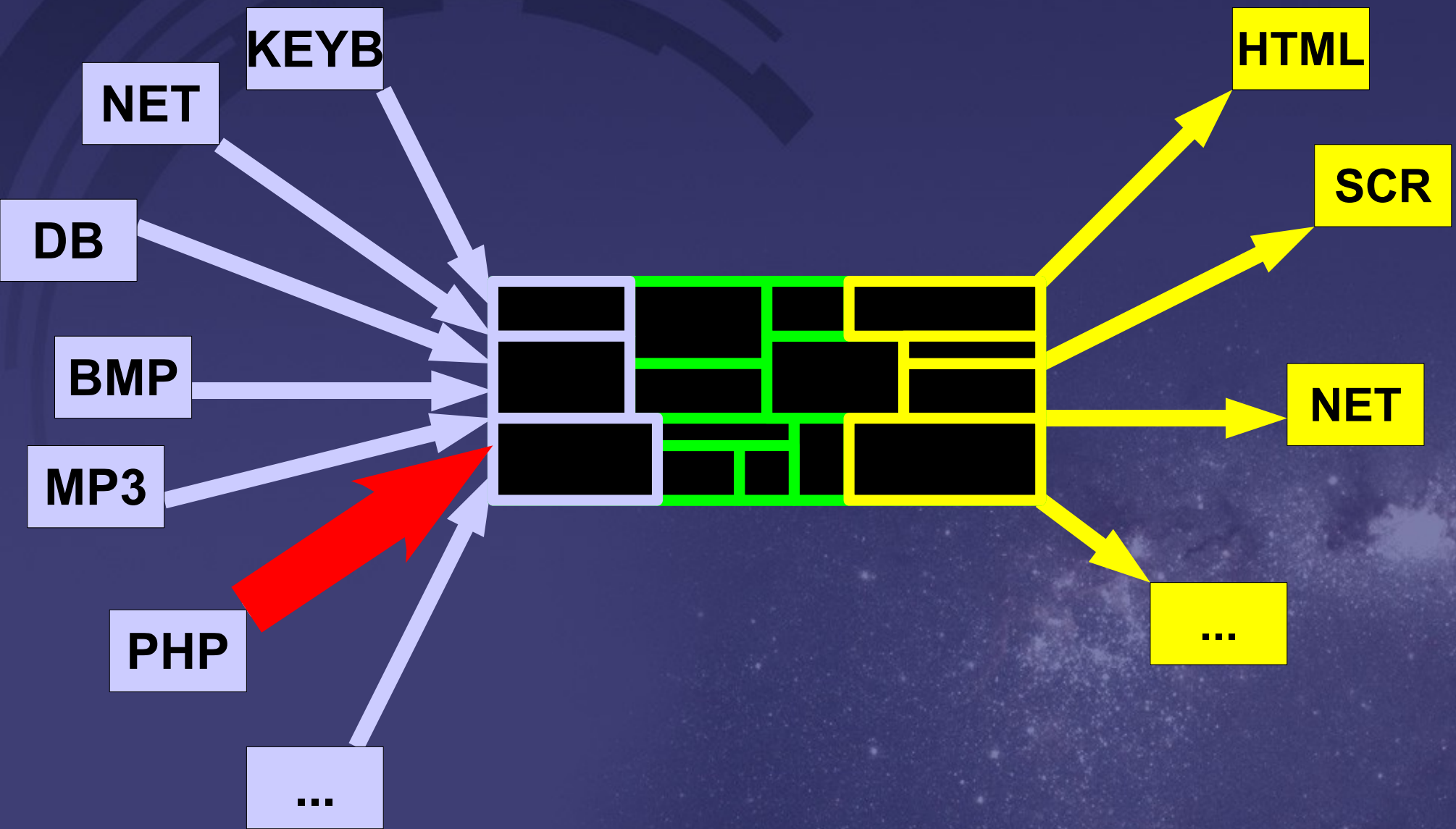
Wszystko co pochodzi z zewnątrz stanowi potencjalne zagrożenie



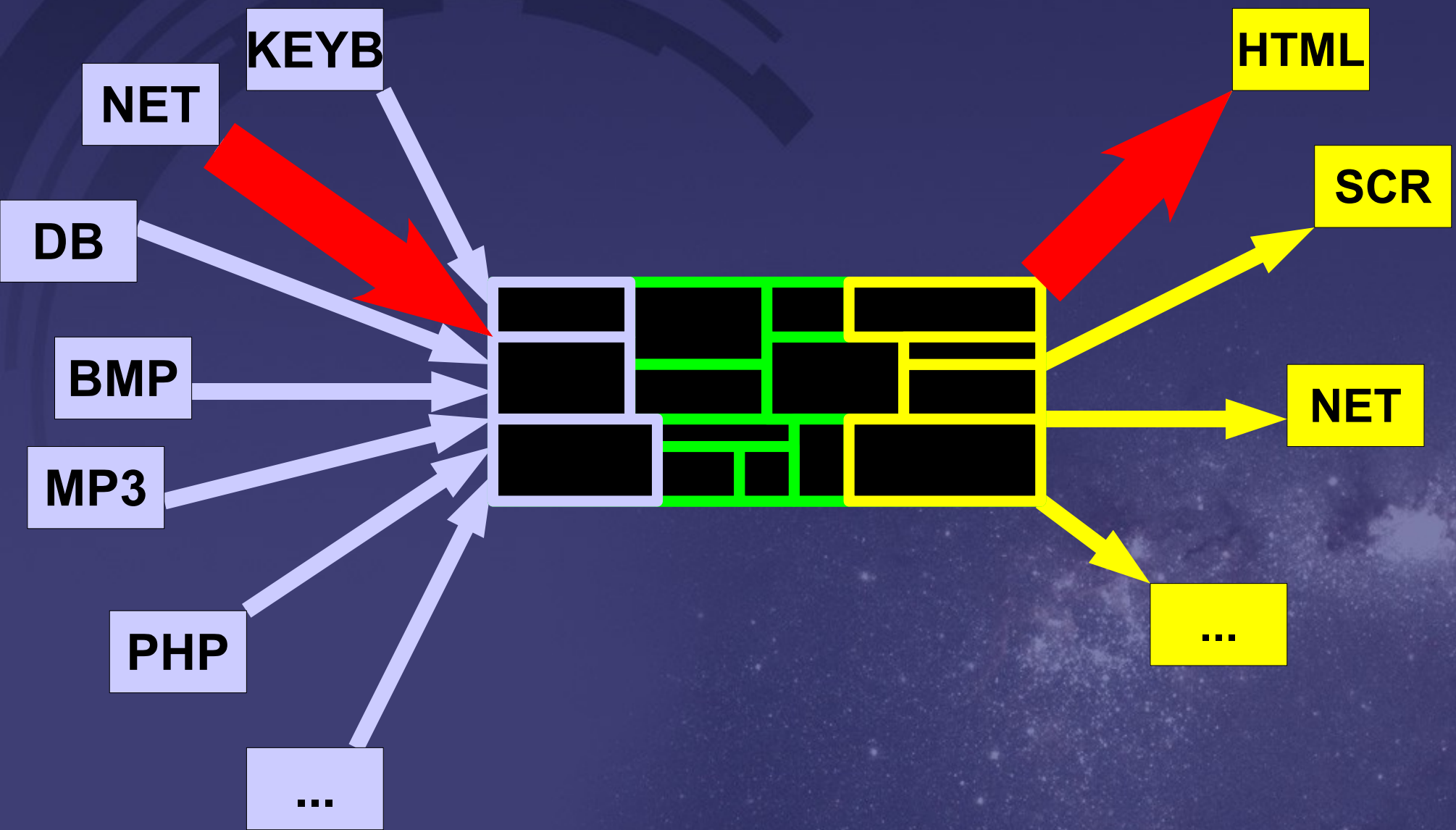
Wszystko co pochodzi z zewnątrz stanowi potencjalne zagrożenie



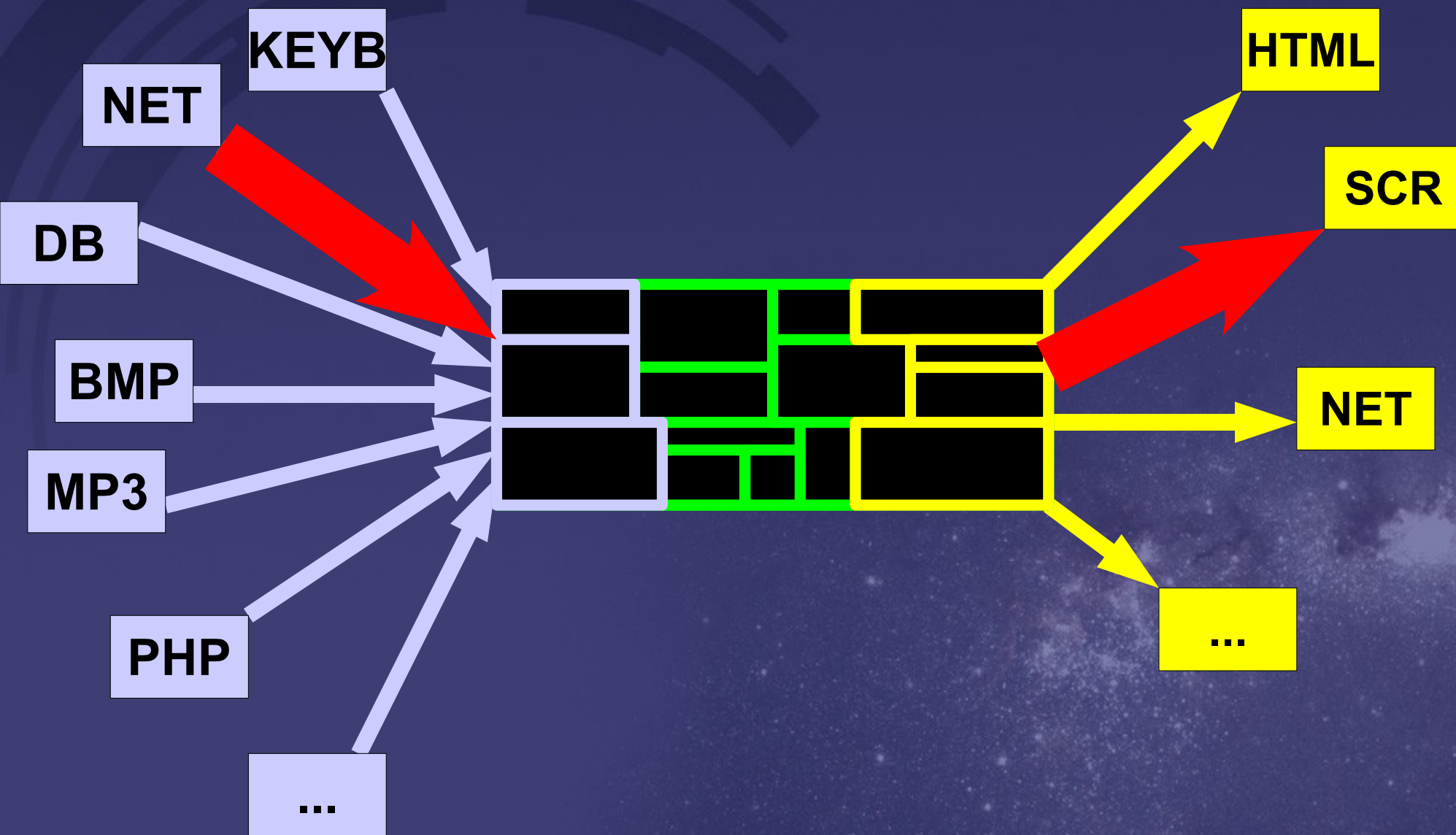
Wszystko co pochodzi z zewnątrz stanowi potencjalne zagrożenie



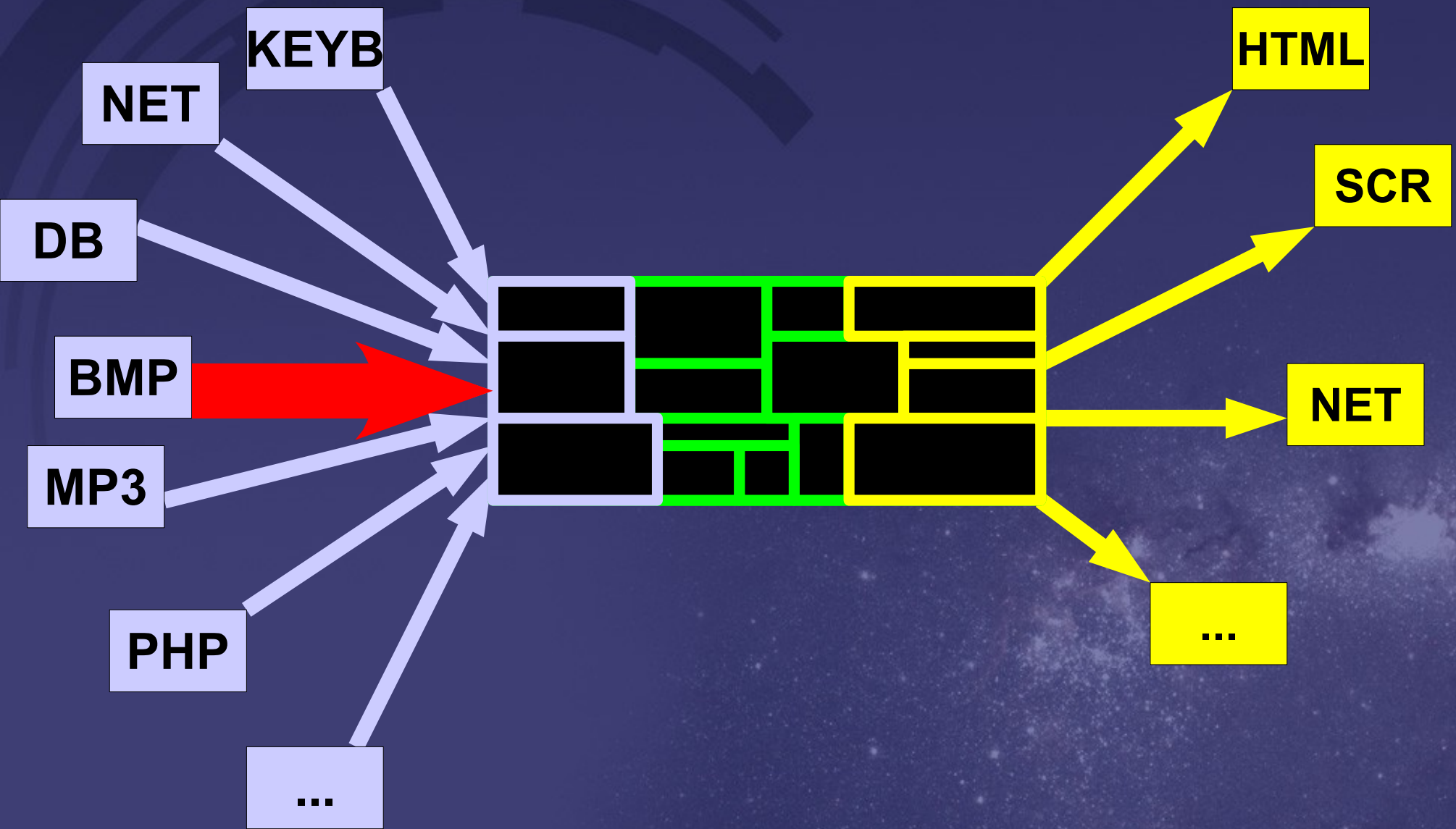
Wszystko co pochodzi z zewnątrz stanowi potencjalne zagrożenie



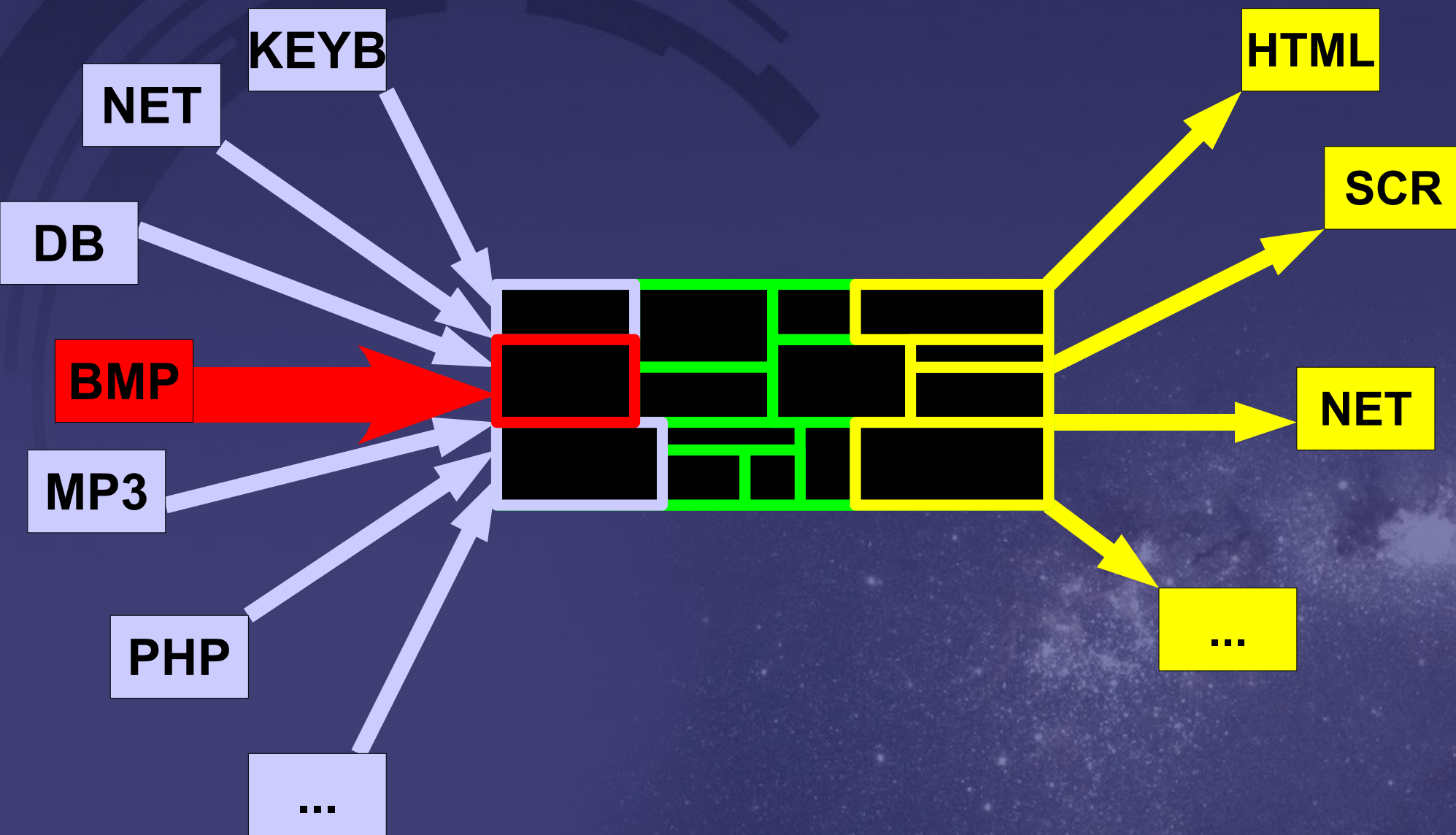
Wszystko co pochodzi z zewnątrz stanowi potencjalne zagrożenie



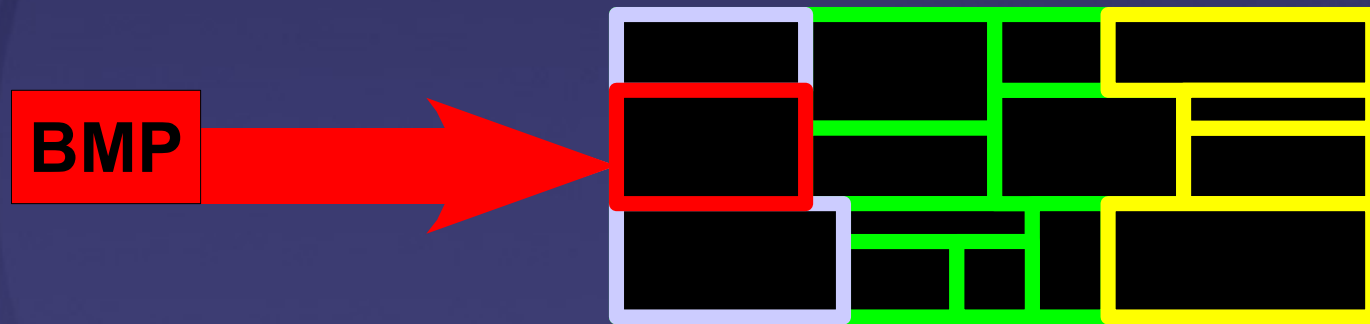
Wszystko co pochodzi z zewnątrz stanowi potencjalne zagrożenie



Jak szukać błędu ?



Jak szukać błędu ?



Jak szukać błędu ?



Jak szukać błędu ?

BMP



```
graph LR; A[BMP] --> B[OBŚŁUGA BMP]
```

**OBŚŁUGA
BMP**



Jak szukać błędu ?

Metoda A: (app oriented)
„Czytając” kod obsługujący format

BMP



The diagram consists of a red rectangular box on the left containing the text 'BMP'. A thick red arrow points from this box to a black rectangular box on the right with a red border, containing the text 'OBSŁUGA BMP' in green. Below the black box is a white triangle pointing upwards towards it.

**OBSŁUGA
BMP**



Jak szukać błędu ?

Metoda B: (data oriented)
Analizując format wejściowy



Jak szukać błędu ?

Zorientowanie na format wejściowy
pozwała naraz testować kilka-kilkanaście
aplikacji



Jak szukać błędu ?

KROK 1

Wybierz format/protokół
który chcesz analizować

Metoda 1: mam ulubioną aplikację...

Metoda 2: format XYZ brzmi fajnie!

Metoda 3: rand()



Jak szukać błędu ?

KROK 2

Zdobądź dokumentacje!

<http://www.wotsit.org>

<http://en.wikipedia.org>

<http://www.faqs.org/rfcs/>

Use google, Luke!



Jak szukać błędu ?

KROK 3

Napisz program tworzący
prawidłowy pakiet/plik

C/C++/Java/Python/etc



Jak szukać błędu ?

KROK 4

Rozpocznij analizę dokumentacji
(o tym za chwilę),
modyfikując stworzony wcześniej
program tak, aby generował
„poprawione” pakiety/pliki



Jak szukać błędu ?

KROK 5

Sprawdź jak zachowują się wybrane aplikacje gdy „zjedzą” „poprawiony” pakiet/plik

(exception monitor, debugger)



Różnice w interpretacji

Programista vs **Bughunter**



Różnice w interpretacji

Programista vs Bughunter

...cytat z dokumentacji...

Co myśli
programista

Co myśli
bughunter



Różnice w interpretacji

Programista vs Bughunter

Bezpośrednio po nagłówku znajdują się dane obrazu...



Różnice w interpretacji

Programista vs Bughunter

Bezpośrednio po nagłówku znajdują się dane obrazu...

Po nagłówku na pewno będą dane obrazu



Różnice w interpretacji

Programista vs Bughunter

Bezpośrednio po nagłówku znajdują się dane obrazu...

Po nagłówku na pewno będą dane obrazu

Utnę plik po nagłówku ;F



Różnice w interpretacji

Programista vs Bughunter

Wartość pola SIZE musi być mniejsza
lub równa 16



Różnice w interpretacji

Programista vs Bughunter

Wartość pola SIZE musi być mniejsza
lub równa 16

Wartość pola
SIZE zawsze
będzie ≤ 16



Różnice w interpretacji

Programista vs Bughunter

Wartość pola SIZE musi być mniejsza
lub równa 16

Wartość pola
SIZE zawsze
będzie ≤ 16

Ustawie SIZE na
55 i zobaczę co
się stanie...



Różnice w interpretacji

Programista vs Bughunter

Pole NR_KOLOR określa ilość kolorów w paletcie



Różnice w interpretacji

Programista vs Bughunter

Pole NR_KOLOR określa ilość kolorów w palecie

Wszystkie numery kolorów w obrazie są \leq NR_KOLOR



Różnice w interpretacji

Programista vs Bughunter

Pole NR_KOLOR określa ilość kolorów w palecie

Wszystkie numery kolorów w obrazie są \leq NR_KOLOR

Powiem że paleta ma 20 kolorów, po czym użyję koloru numer 55 w obrazie!



Różnice w interpretacji

Programista vs Bughunter

Chunk PALETA zawiera paletę kolorów dla całego obrazu



Różnice w interpretacji

Programista vs Bughunter

Chunk PALETA zawiera paletę kolorów dla całego obrazu

Chunk PALETA wystąpi tylko raz, bo nikt nie używa więcej niż jednej palety



Różnice w interpretacji

Programista vs Bughunter

Chunk PALETA zawiera paletę kolorów dla całego obrazu

Chunk PALETA wystąpi tylko raz, bo nikt nie używa więcej niż jednej palety

Wrzucę 10 palet!



Różnice w interpretacji

Programista vs Bughunter

Chunk PALETA zawiera paletę kolorów dla całego obrazu

Chunk PALETA wystąpi tylko raz, bo nikt nie używa więcej niż jednej palety

Wrzucę 10 palet!
Albo żadnej!



Różnice w interpretacji

Programista vs Bughunter

Implementacja powinna sprawdzać
czy XYZ jest poprawne



Różnice w interpretacji

Programista vs Bughunter

Implementacja powinna sprawdzać
czy XYZ jest poprawne

Nie chcę mi się...



Różnice w interpretacji

Programista vs Bughunter

Implementacja powinna sprawdzać
czy XYZ jest poprawne

Nie chcę mi się...

Ciekawe czy
programiście
się chciało...



Różnice w interpretacji

Programista vs Bughunter

Pole OFFSET zawiera
offset danych w pliku



Różnice w interpretacji

Programista vs Bughunter

Pole OFFSET zawiera
offset danych w pliku

Wartość OFFSET
pokazuje gdzieś
w środku pliku na
dane



Różnice w interpretacji

Programista vs Bughunter

Pole OFFSET zawiera
offset danych w pliku

Wartość OFFSET
pokazuje gdzieś
w środku pliku na
dane

Ustawie OFFSET
ujemne lub większe
od wielkości pliku



Różnice w interpretacji

Programista vs Bughunter

Każdy chunk ma pole NEXT które mówi gdzie jest następny chunk



Różnice w interpretacji

Programista vs Bughunter

Każdy chunk ma pole NEXT które mówi gdzie jest następny chunk

O! lista chunków!
Przejdę sobie
po niej w pętli



Różnice w interpretacji

Programista vs Bughunter

Każdy chunk ma pole NEXT które mówi gdzie jest następny chunk

O! lista chunków!
Przejdę sobie
po niej w pętli

O! Zapętłona lista
chunków! Niech
któryś wskazuje
sam na siebie!



Różnice w interpretacji - podsumowanie

Programista vs Bughunter

ZDANIE



Różnice w interpretacji - podsumowanie

Programista vs Bughunter

ZDANIE

ZDANIE na pewno
jest prawdziwe
i wszyscy wg
niego tworzą
pliki / pakiety!

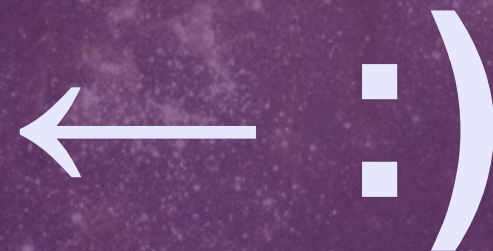


Różnice w interpretacji - podsumowanie

Programista vs Bughunter

ZDANIE

ZDANIE na pewno
jest prawdziwe
i wszyscy wg
niego tworzą
pliki / pakiety!



Różnice w interpretacji - podsumowanie

**Co jest złe?
Formaty czy implementacje?**



Enter teh BMP

Case study - BMP



Enter teh BMP

BMP w skrócie:

- bardzo prosty format graficzny
- brak kompresji lub kompresja RLE
- znany z duużych plików
- znany z Windowsa :)
- obsługuje do 8 bitów z paletą
- obsługuje od 16 do 32 bitów RGB



Enter teh BMP

Budowa BMP

BITMAPFILEHEADER

BITMAPINFOHEADER

PALETTE

IMAGE DATA



Bitmap File Header

BITMAPFILEHEADER

BITMAPINFOHEADER

PALETTE

IMAGE DATA

```
UINT    bfType;  
DWORD   bfSize;  
UINT    bfReserved1;  
UINT    bfReserved2;  
DWORD   bfOffBits;
```



Bitmap File Header

BITMAPFILEHEADER

BITMAPINFOHEADER

PALETTE

IMAGE DATA

```
UINT    bfType;  
DWORD  bfSize;  
UINT    bfReserved1;  
UINT    bfReserved2;  
DWORD  bfOffBits;
```

Specifies the size of the file, in bytes.



Bitmap File Header

BITMAPFILEHEADER

BITMAPINFOHEADER

PALETTE

IMAGE DATA

```
UINT    bfType;  
DWORD   bfSize;  
UINT    bfReserved1;  
UINT    bfReserved2;  
DWORD  bfOffBits;
```

Specifies the byte offset from the **BITMAPFILEHEADER** structure to the actual bitmap data in the file.
(demo 001 002)



Bitmap File Header

BITMAPFILEHEADER

BITMAPINFOHEADER

PALETTE

IMAGE DATA

```
DWORD biSize; (1 of 2)
LONG biWidth;
LONG biHeight;
WORD biPlanes;
WORD biBitCount;
DWORD biCompression;
```



Bitmap File Header

BITMAPFILEHEADER

BITMAPINFOHEADER

PALETTE

IMAGE DATA

DWORD **biSize;** (1 of 2)

LONG **biWidth;**

LONG **biHeight;**

WORD **biPlanes;**

WORD **biBitCount;**

DWORD **biCompression;**

Specifies the width/height
of the bitmap, in pixels.

$(SZ = biWidth * biHeight * biBitCount/8)$

(memory DoS, Int Overflow, Sig/UnSig)



Bitmap File Header

BITMAPFILEHEADER

BITMAPINFOHEADER

PALETTE

IMAGE DATA

DWORD **biSize;** (1 of 2)

LONG **biWidth;**

LONG **biHeight;**

WORD biPlanes;

WORD **biBitCount;**

DWORD **biCompression;**

Specifies the number of planes for the target device. This member must be set to 1.



Bitmap File Header

BITMAPFILEHEADER

BITMAPINFOHEADER

PALETTE

IMAGE DATA

```
DWORD biSize; (1 of 2)
LONG biWidth;
LONG biHeight;
WORD biPlanes;
WORD biBitCount;
DWORD biCompression;
```

Specifies the number of bits per pixel.
This value **must be** 1, 4, 8, or 24.
(32 bits?)



Bitmap File Header

BITMAPFILEHEADER

BITMAPINFOHEADER

PALETTE

IMAGE DATA

(2 of 2)

```
DWORD biSizeImage;  
LONG biXPelsPerMeter;  
LONG biYPelsPerMeter;  
DWORD biClrUsed;  
DWORD biClrImportant;
```



Bitmap File Header

BITMAPFILEHEADER

BITMAPINFOHEADER

PALETTE

IMAGE DATA

(2 of 2)

```
DWORD biSizeImage;  
LONG biXPelsPerMeter;  
LONG biYPelsPerMeter;  
DWORD biClrUsed;  
DWORD biClrImportant;
```

Specifies the number of color indexes in the color table actually used by the bitmap. [...] the `biClrUsed` member must be set to zero or to the actual size of the color table.



Bitmap File Header

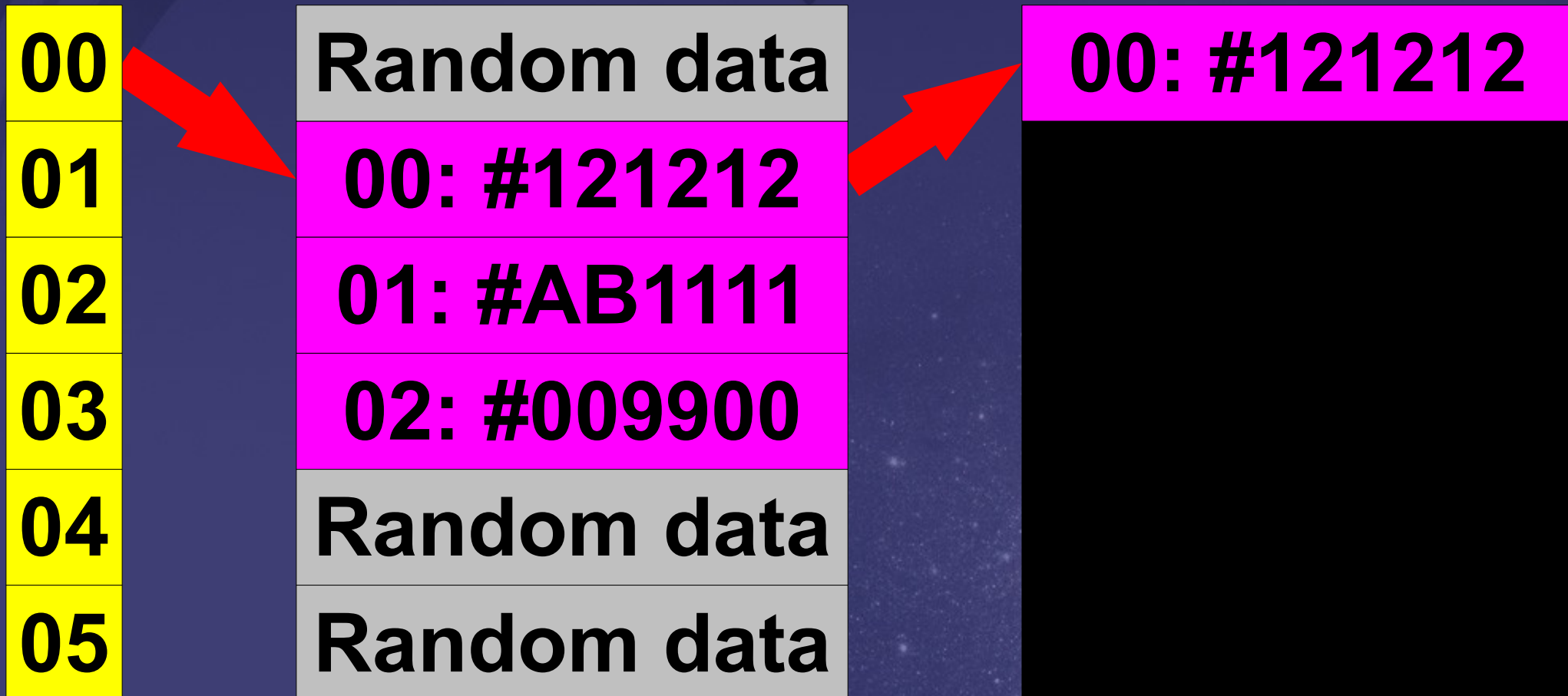
biClrUsed = 3

00	Random data
01	00: #121212
02	01: #AB1111
03	02: #009900
04	Random data
05	Random data



Bitmap File Header

biClrUsed = 3



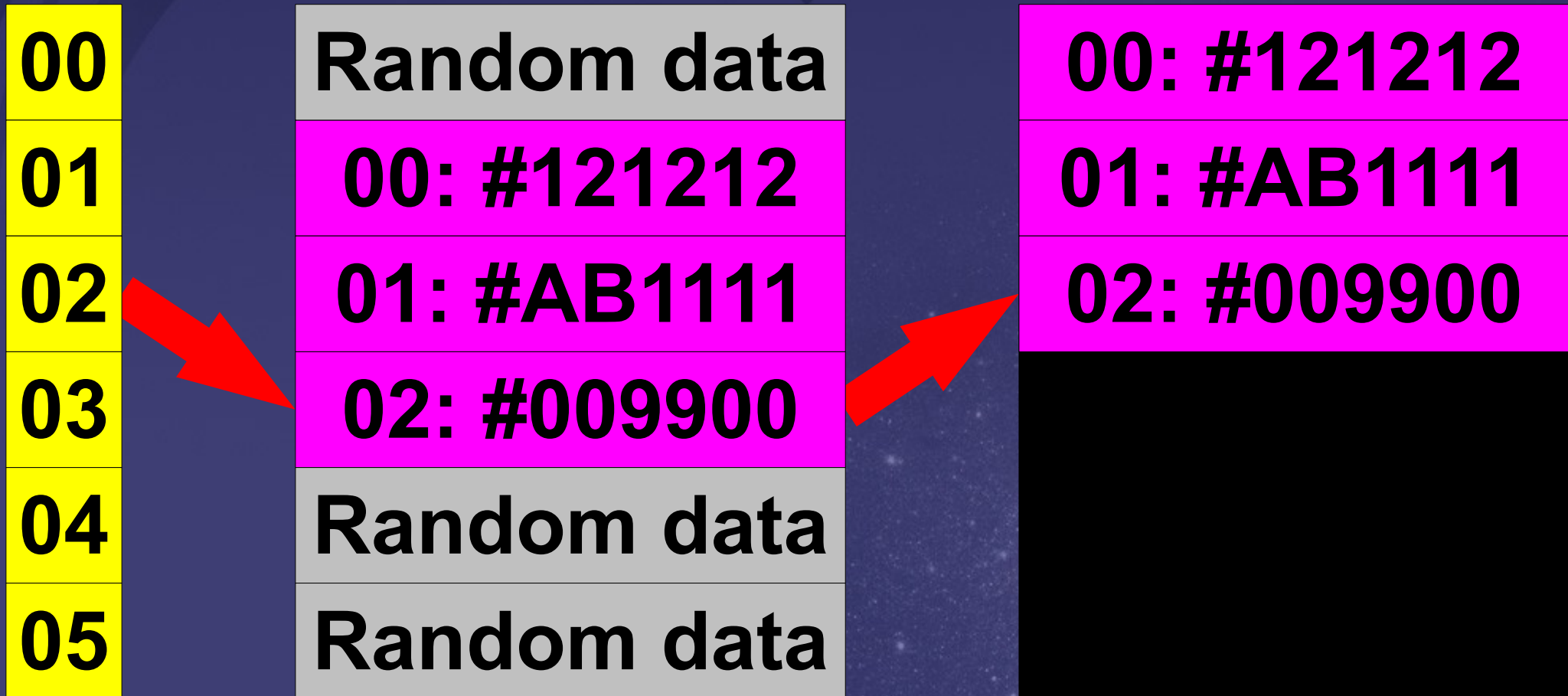
Bitmap File Header

biClrUsed = 3



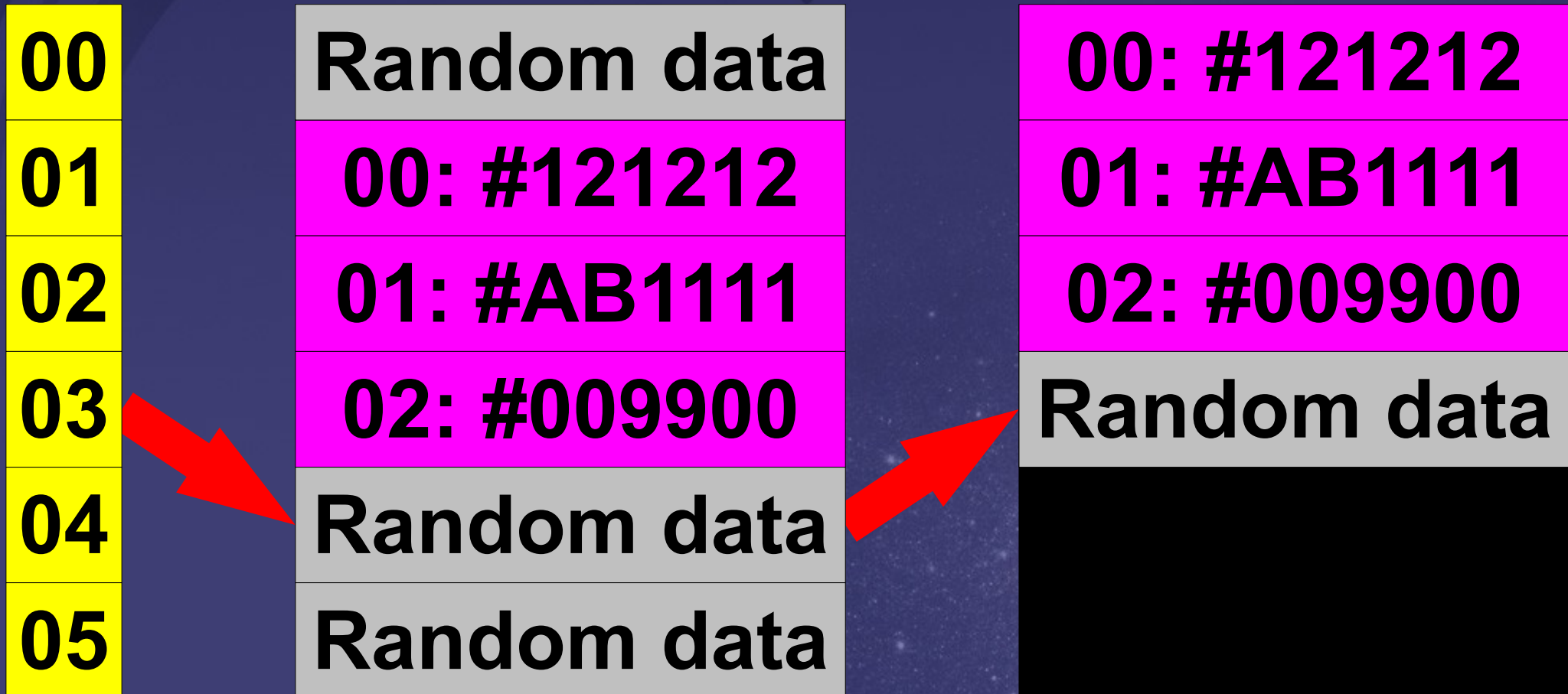
Bitmap File Header

biClrUsed = 3



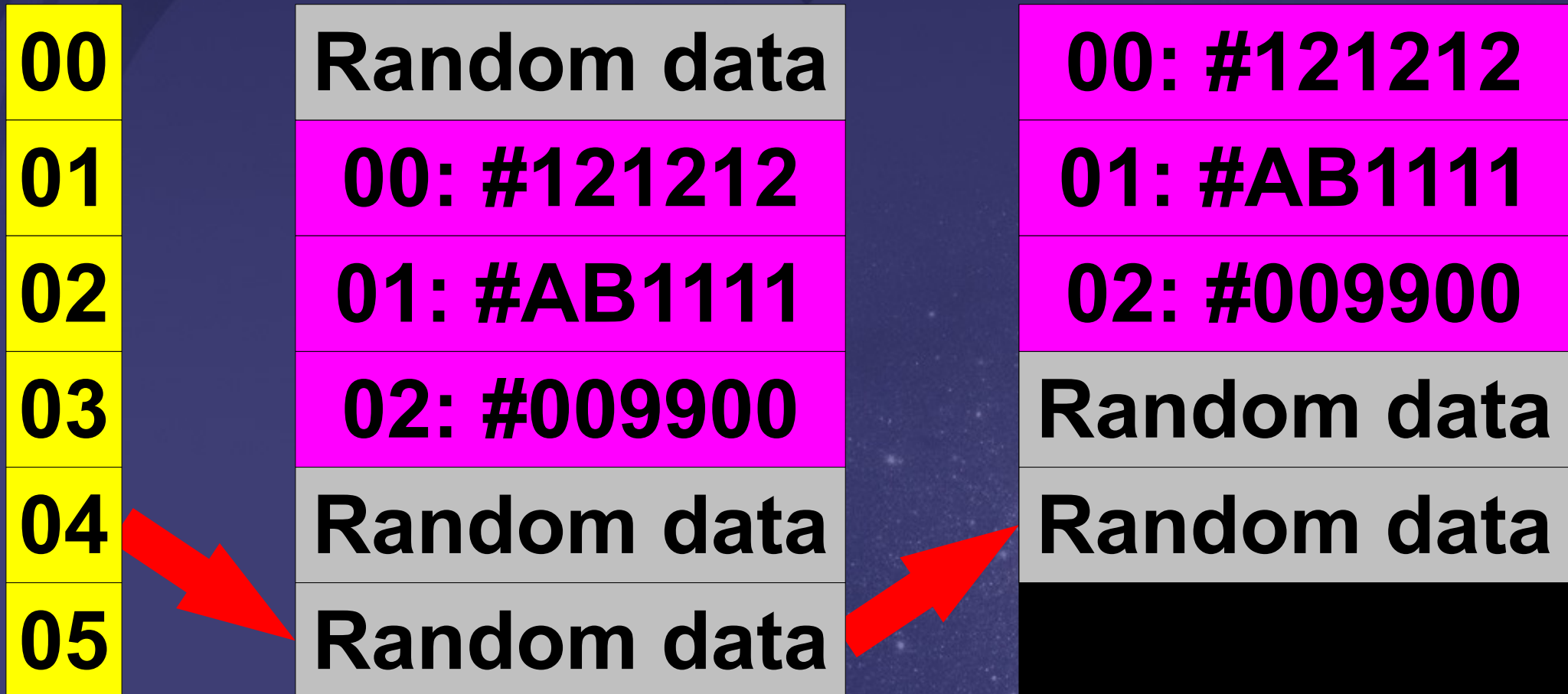
Bitmap File Header

biClrUsed = 3



Bitmap File Header

biClrUsed = 3



Bitmap File Header

Demo 003



RLE

BITMAPFILEHEADER

BITMAPINFOHEADER

PALETTE

IMAGE DATA

Run Length Encoding:
biCompression = BI_RLE8
biCompression = BI_RLE4

...



RLE

BITMAPFILEHEADER

BITMAPINFOHEADER

PALETTE

IMAGE DATA

Run Length Encoding:
biCompression = BI_RLE8
biCompression = BI_RLE4

AAAAAAAAAAABBBBCCCCCABCABCAAAA



RLE

BITMAPFILEHEADER

BITMAPINFOHEADER

PALETTE

IMAGE DATA

Run Length Encoding:
biCompression = BI_RLE8
biCompression = BI_RLE4

AAAAAAAAAAAA BBBBCCCCC ABCABCAAAAA

10 'A'



RLE

BITMAPFILEHEADER

BITMAPINFOHEADER

PALETTE

IMAGE DATA

Run Length Encoding:
biCompression = BI_RLE8
biCompression = BI_RLE4

AAAAAAAAAA**BBBB**CCCCCABCABCAAAA

10 'A' **04 'B'**



RLE

BITMAPFILEHEADER

BITMAPINFOHEADER

PALETTE

IMAGE DATA

Run Length Encoding:
biCompression = BI_RLE8
biCompression = BI_RLE4

AAAAAAAAAAABBBB**CCCCC**ABCABCAAAAA

10 'A' 04 'B' **05 'C'**



RLE

BITMAPFILEHEADER

BITMAPINFOHEADER

PALETTE

IMAGE DATA

Run Length Encoding:
biCompression = BI_RLE8
biCompression = BI_RLE4

AAAAAAAAAAABBBBCCCCC**ABCABC**AAAAA

10 'A' 04 'B' 05 'C' **00 06 'ABCABC'**



RLE

BITMAPFILEHEADER

BITMAPINFOHEADER

PALETTE

IMAGE DATA

Run Length Encoding:
biCompression = BI_RLE8
biCompression = BI_RLE4

AAAAAAAAAAABBBBCCCCCABCABC**AAAAA**

10 'A' 04 'B' 05 'C' 00 06 'ABCABC' **05 'A'**



RLE

BITMAPFILEHEADER

BITMAPINFOHEADER

PALETTE

IMAGE DATA

Run Length Encoding:
biCompression = BI_RLE8
biCompression = BI_RLE4

AAAAAAAAAAABBBBCCCCCABCABCAAAA

10 'A' 04 'B' 05 'C' 00 06 'ABCABC' 05 'A' 00 01



RLE

BITMAPFILEHEADER

BITMAPINFOHEADER

PALETTE

IMAGE DATA

Run Length Encoding:

`biCompression = BI_RLE8`

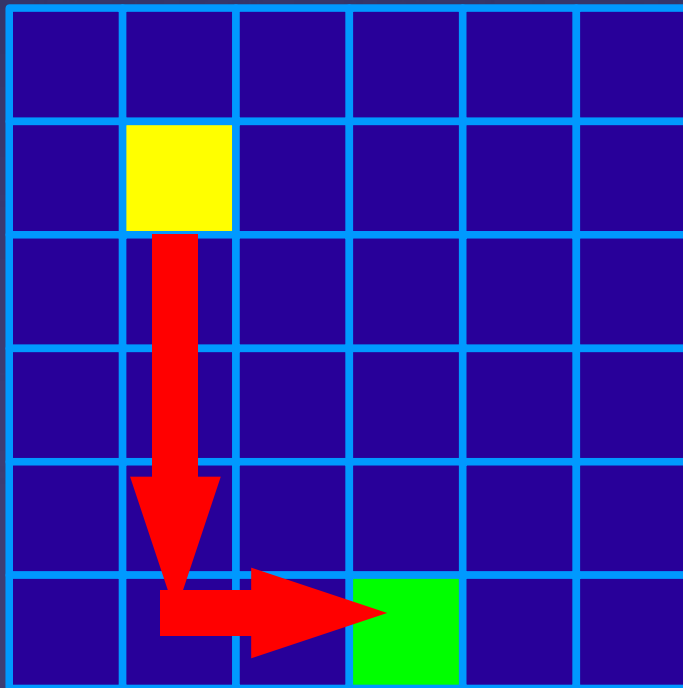
`biCompression = BI_RLE4`

`00 02 XX YY`: Delta. The two bytes following the escape contain unsigned values indicating the horizontal and vertical offsets of the next pixel from the current position.



RLE

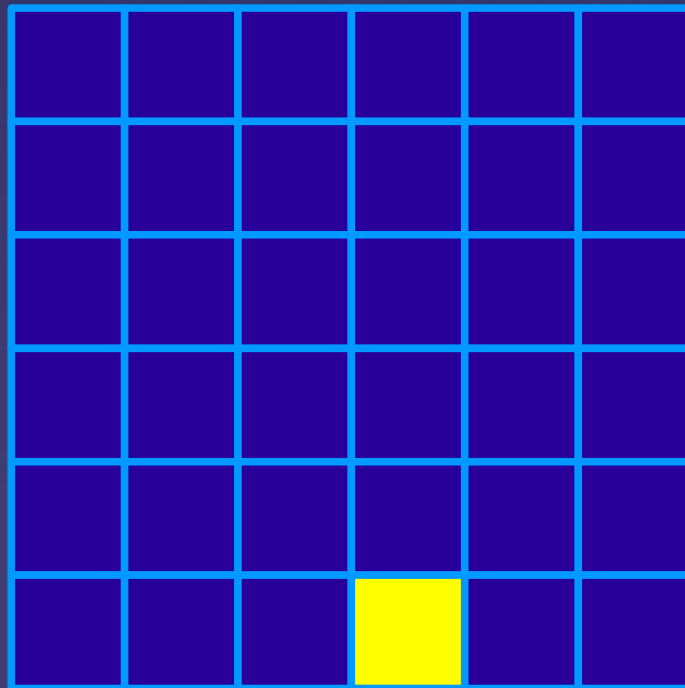
00 02 02 04



RLE

00 02 02 04

00 02 02 04



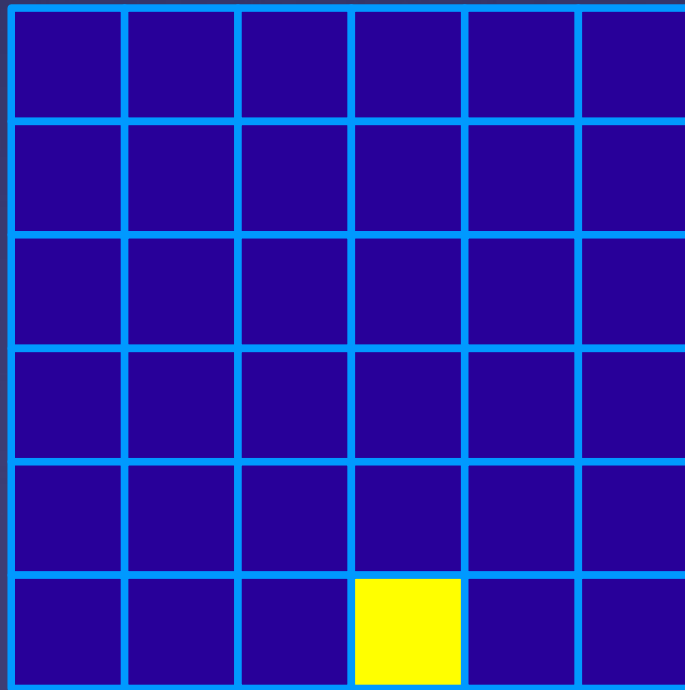
???



RLE

00 02 02 04

00 02 02 04



Demo 004



BMP podsumowanie

...



Enter teh GIF

Case study - GIF



Enter teh GIF

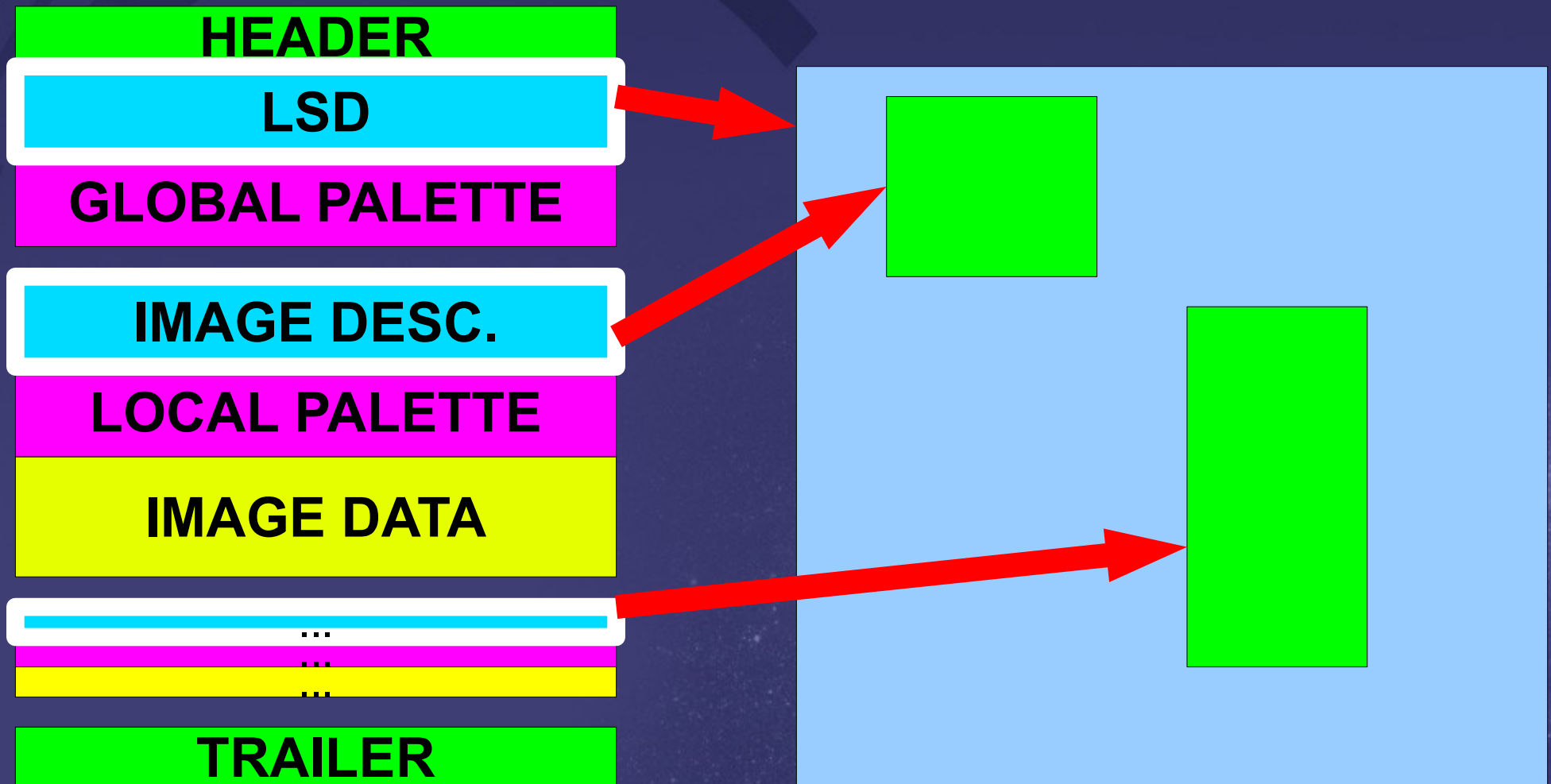
Krótko o GIF:

- bezstratna kompresja (LZW, ?)
- 8 bitów, paleta kolorów (>256)
- animacje! (WEB 1.0 ;>)
- oparty o chunki
- logical screen vs image(s)
- GIF87 vs GIF89



Enter teh GIF

Budowa GIF



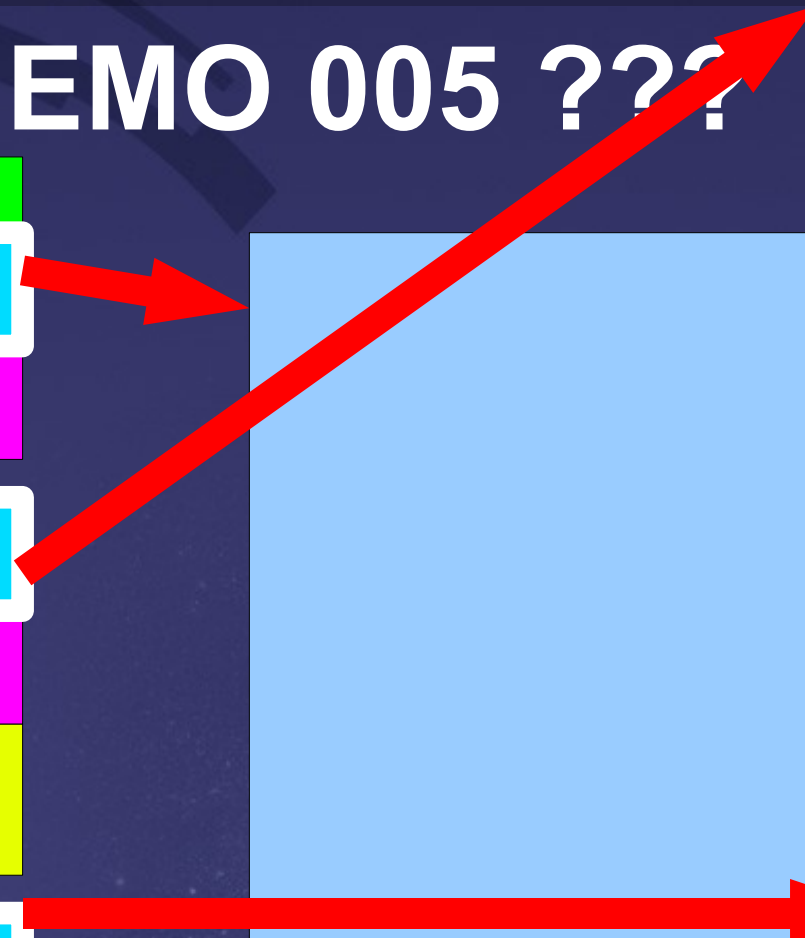
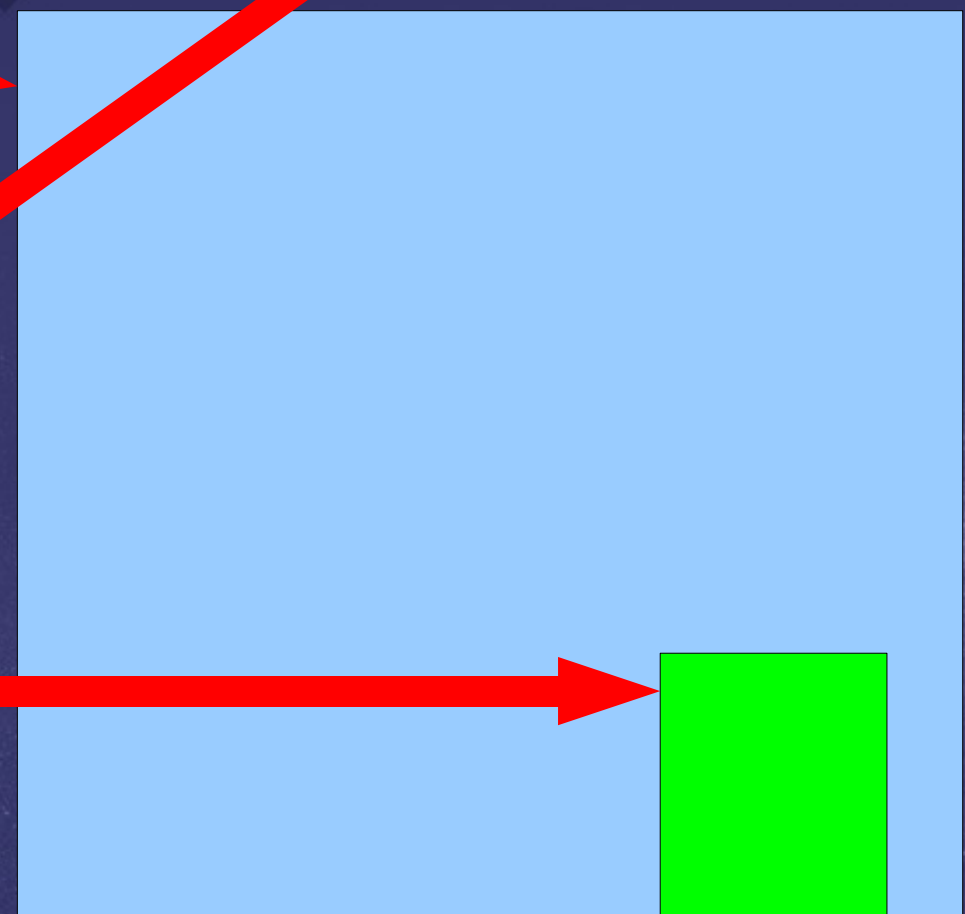
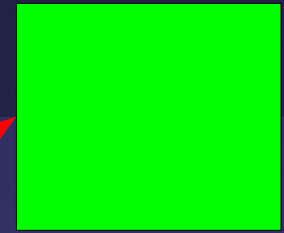
Enter teh GIF

??? Budowa GIF ???



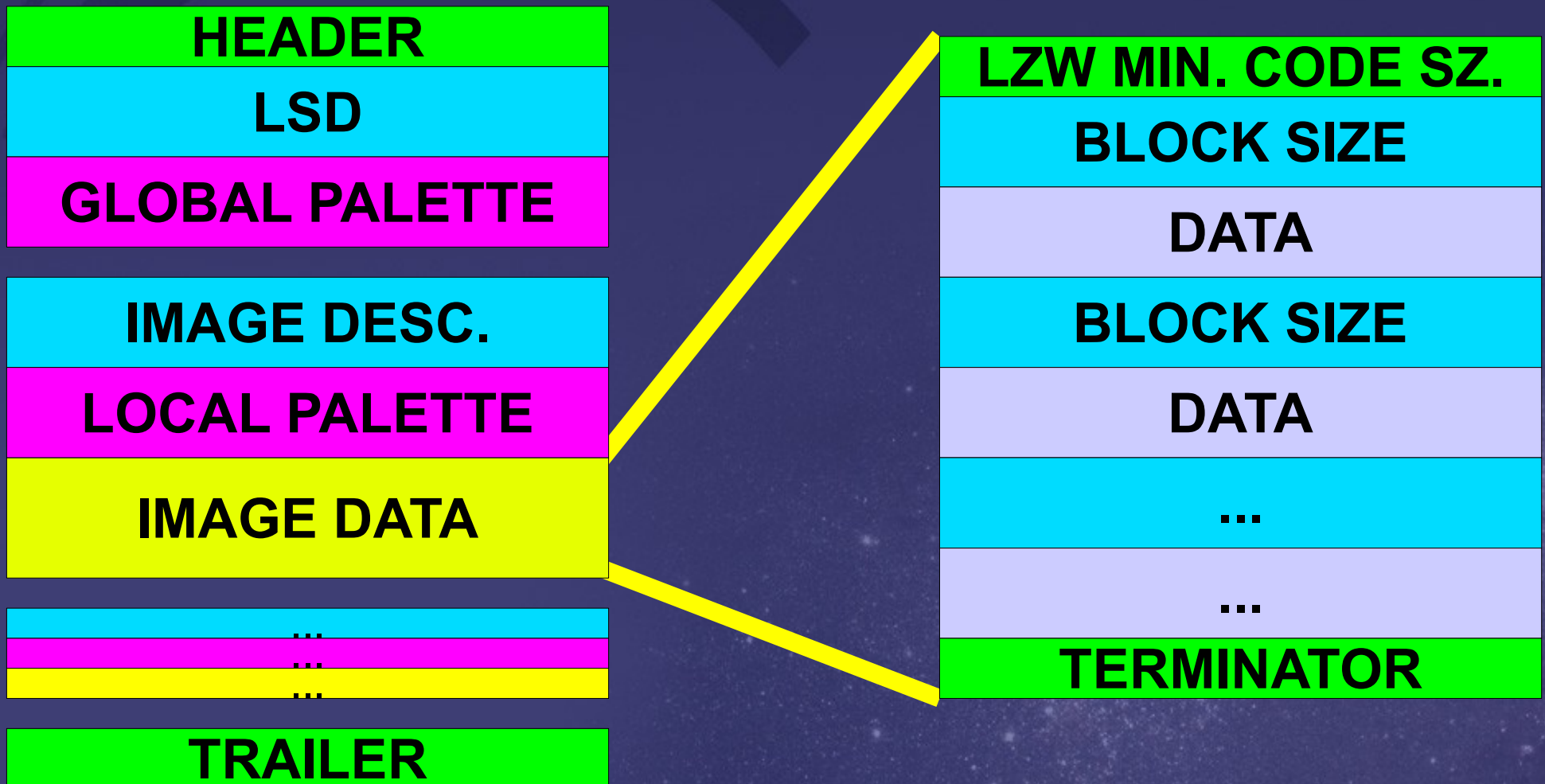
Enter teh GIF

??? DEMO 005 ???



Enter teh GIF

Budowa GIF



Enter teh GIF

Budowa GIF

The output codes are of variable length, starting at $\langle \text{code size} \rangle + 1$ bits per code, **up to 12 bits** per code. This defines a maximum code value of 4095 (hex FFF).



Enter teh GIF SDL_Image

```
...  
unsigned char c; ← code size  
...  
LWZReadByte(src, TRUE, c)
```



Enter teh GIF

SDL_Image

```
LWZReadByte(..., int input_code_size)
...
static int table[2][(1 << MAX_LWZ_BITS)];
...
set_code_size = input_code_size;
...
clear_code = 1 << set_code_size;
...
for (i = 0; i < clear_code; ++i) {
    table[0][i] = 0;
    table[1][i] = i;
}
```



Enter teh GIF

SDL_Image – DEMO 006



Escape teh GIF

GIF - podsumowanie



Inne – FTP, RAR, ZIP

FTP – Nazwy plików w listingu

LIST (LIST)

If the pathname specifies a directory or other group of files, the server should transfer a **list of files** in the specified directory.



Inne – FTP, RAR, ZIP

FTP – Nazwy plików w listingu

LIST (LIST)

If the pathname specifies a directory or other group of files, the server should transfer a **list of files** in the specified directory.

DEMO 007!



Inne – FTP, RAR, ZIP

RAR, ZIP, etc...

**(RAR) File name - string of
NAME_SIZE bytes size**



Inne – FTP, RAR, ZIP

RAR, ZIP, etc...

(ZIP)

[local file header 1]

VS

[central directory]



EOF

Podsumowanie...



The End

Dziękuję za uwagę!
Czas na pytania ;>

```
--(* e-mail *)--  
gynvael@coldwind.pl  
michael@hispasec.com
```

```
--(* blog *)--  
http://gynvael.coldwind.pl
```

